Pacific Graphics 2024
R. Chen, T. Ritschel, and E. Whiting
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 7

NeuPreSS: Compact Neural Precomputed Subsurface Scattering for Distant Lighting of Heterogeneous Translucent Objects

T. TG[1], J. R. Frisvad[1], R. Ramamoorthi[2], and H. W. Jensen[3]

[1]Technical University of Denmark, Denmark
[2]University of California, San Diego, USA
[3]Keyshot, USA

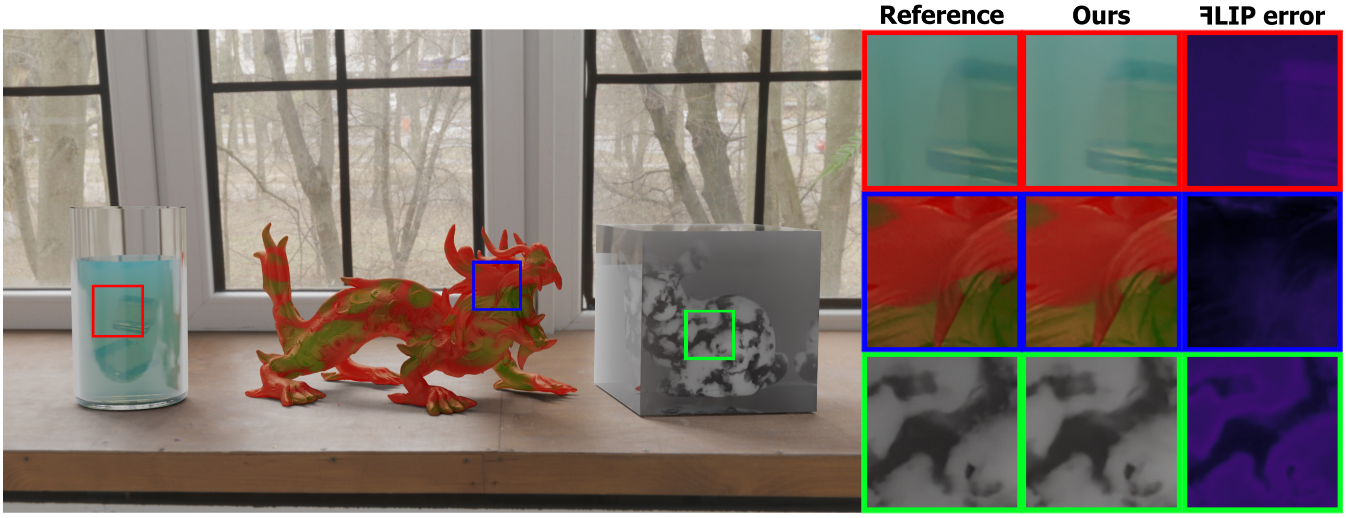


Figure 1: *Three objects exhibiting heterogeneous subsurface scattering: a liquid containing ice cubes, the XYZ RGB dragon with two materials distributed using a noise function, and a paper weight with a marble bunny inside. The objects are illuminated by an environment map and holdout geometry is used to catch shadows. All the models in this scene illustrate complex light transport behind a refractive interface with clearly perceptible heterogeneity. Despite our subsurface scattering being limited to distant lighting, the outer surface reflection is not and our neural precomputed method demonstrates high accuracy as compared with the reference. After training our model for each object using directional lights, we render using the outer object geometry only. Knowledge of internal geometries is then no longer needed.*

Abstract

Monte Carlo rendering of translucent objects with heterogeneous scattering properties is often expensive both in terms of memory and computation. If the scattering properties are described by a 3D texture, memory consumption is high. If we do path tracing and use a high dynamic range lighting environment, the computational cost of the rendering can easily become significant. We propose a compact and efficient neural method for representing and rendering the appearance of heterogeneous translucent objects. Instead of assuming only surface variation of optical properties, our method represents the appearance of a full object taking its geometry and volumetric heterogeneities into account. This is similar to a neural radiance field, but our representation works for an arbitrary distant lighting environment. In a sense, we present a version of neural precomputed radiance transfer that captures relighting of heterogeneous translucent objects. We use a multi-layer perceptron (MLP) with skip connections to represent the appearance of an object as a function of spatial position, direction of observation, and direction of incidence. The latter is considered a directional light incident across the entire non-self-shadowed part of the object. We demonstrate the ability of our method to compactly store highly complex materials while having high accuracy when comparing to reference images of the represented object in unseen lighting environments. As compared with path tracing of a heterogeneous light scattering volume behind a refractive interface, our method more easily enables importance sampling of the directions of incidence and can be integrated into existing rendering frameworks while achieving interactive frame rates.

CCS Concepts

• ***Computing methodologies*** → ***Reflectance modeling; Neural networks;***

1. Introduction

Many rendering techniques do not capture the bleeding of light through a translucent object. Surface scattering models like those based on the bidirectional reflectance distribution function (BRDF) are local and cannot capture the effect; the models by Fan et al. [FWH*22] and Zeltner et al. [ZRW*23] are recent examples. Volume path tracing is needed to capture how light entering the object at one surface location scatters and emerges at another. A bidirectional texture function (BTF) [RGJW20, KMX*21] is able to capture some subsurface scattering because it includes two spatial dimensions for surface texturing. A BTF is however meant to model a surface patch and not the global bleeding of light through an object. The bidirectional scattering-surface reflectance distribution function (BSSRDF) includes the full scattering of light incident at one location to light emerging in another object surface location. This function captures the same effects as full volume path tracing except that we cannot place the camera inside the object. Volume path tracing is a local formulation where we model the light transport locally one path at the time; the BSSRDF is a global formulation considering the full subsurface scattering (all paths) between positions and directions of incidence and emergence [Pre65].

When using a BSSRDF, it is most common to employ an approximate analytic model [JMLH01]. Models exist that generalize to include dependency on the directions of incidence and observation [FHK14, FD17] or on the local object geometry if a specific sampling technique is employed [VKJ19]. Alternatively, one can speed up full volume path tracing by learning the scattering in a sphere [LHW21]. These methods can deal with local lighting but are dedicated to homogeneous volumes. Our method, on the other hand, can evaluate the subsurface light transport with full heterogeneity, geometry, and directionality and can be used with different sampling techniques but assumes distant lighting.

Rendering of a heterogeneous translucent object is computationally expensive because the scattering events happen behind a refractive interface. This means that rays cannot be traced directly toward the light making importance sampling more difficult when volume path tracing is used for rendering [DWWH20]. An often used solution is to approximate the subsurface scattering using a factored BSSRDF representation [PvBM*06, CPZT12, Kur21] or by solving the diffusion equation for the object volume [WZT*08, WWH*10, AWB11]. These approaches however rely on approximate representations of the surface and subsurface scattering.

An alternative way to capture the appearance of a translucent object is by precomputed radiance transfer (PRT) [SLS05, BDBS22], but PRT methods assume low-frequency variation of the radiance as a function of the directions of incidence and observation. Our method is conceptually similar to PRT, but we provide a neural representation of the subsurface scattering that adapts to an arbitrary distant lighting environment which is not necessarily of low frequency. We also do not assume that the subsurface scattering leads to diffuse emergent light, which is a common assumption even in all-frequency PRT models as they are usually based on an analytic BSSRDF model [WTL05].

We represent the appearance of a heterogeneous translucent object by assuming known surface geometry and distant illumination. We care only about light emerging at the surface of the object and therefore use the global BSSRDF representation of the appearance. Our representation depends on the full object geometry, not only one point of incidence. We thus train for each object separately and obtain a method capable of capturing the appearance of a heterogeneous translucent object from an arbitrary view and illuminated by an arbitrary distant lighting environment. Our use of the global formulation makes it easier for us to importance sample the incident illumination as we do not need to evaluate the light reaching scattering events behind a refractive interface. We demonstrate good rendering performance both in terms of accuracy and rendering time as compared with volume path tracing. An example of our model's ability to represent the appearance captured by volume path tracing is in Fig. 1. We compare the accuracy of our model with that of classic appearance representation in spherical harmonics (SH) as well as factored NeRF [ZSD*21], a neural BTF model [KMX*21], and analytic BSSRDF models fitted using differentiable rendering [DLW*22]. Our key contributions:

- A translucent object appearance representation for distant lighting not limited to highly scattering materials, i.e., features behind a refractive interface are faithfully represented despite considering the surface only (Fig. 1, Sec. 3.1).
- A small neural network attaining a very low memory footprint (<3.77MB) despite representing non-trivial subsurface light transport while achieving a significant speedup for rendering compared to path tracing (Table 1).
- An on-surface importance sampling method for our representation along with an online learning methodology for training while precomputing the subsurface scattering (Sec. 3.2).
- Our low-dimensional representation can be used in a PRT context to extract spherical harmonics coefficients for real-time rendering of heterogeneous translucent objects (Sec. 4.3).

Code and data for this paper are available at https://lab.compute.dtu.dk/thtg/NeuPreSS to facilitate replication.

2. Related Work

Full object appearance representation is not uncommon. A popular approach is use of a neural radiance field (NeRF) [MST*21]. This is however a representation of the object appearance in one particular lighting environment. Our method is thus related to techniques for representation and rendering of heterogeneous volumes and in particular relightable NeRF and neural PRT methods.

Relightable NeRF One way to make a NeRF model relightable is to jointly optimize a model for shape, spatially varying BRDF, and illumination [BBJ*21, ZSD*21, LTL*22, ZSH*22, YZL*22]. By assuming a BRDF model, these methods however sacrifice their ability to represent the light transport in a translucent object since the model considers only light incident in the point of observation. Other relightable NeRF models specifically assume a low-frequency lighting environment [SLB*21] or an outdoor sun and sky lighting model [LGF*22]. Gao et al. [GCD*20] and Zeng et al. [ZCD*23] include subsurface scattering but details behind a refractive interface would likely be difficult to capture due to the use of rendered radiance cues/hints. Zhu et al. [ZSB*23] present a radiance field (volume rendering) version of what we propose. However, they do not separate out specular effects (when an object is relit by an environment, the environment is not reflected in the surface), and they ignore refraction. Not accounting for refraction in a volumetric method can lead to issues when the heterogeneous interior of an object exhibits details [BMF*22].

Neural PRT Our concept is a variation of neural PRT [RBRD22, RXL*23], but we do not separate our model into a full environment encoded in spherical harmonics or wavelets and a transport MLP which are combined using a learned operator. Instead, our model is designed for rendering of a translucent object in arbitrary environments, and we include the direction of a distant source directly as an input for the network. This eases the integration of our model into rendering of the represented translucent object in a scene where we can assume that sources of incident illumination are distant. Zheng et al. [ZHM*23] provide a neural PRT-like representation of the dynamic light transport between an object and the surrounding environment but include no example involving a translucent object. Modifying this technique to incorporate subsurface scattering would be a research project in its own right as the described scene representation and its use for estimating indirect illumination does not consider light transfer through objects.

Heterogeneous volumes Goesele et al. [GLL*04] present a method for capturing the appearance of translucent objects. This method has very high storage cost (>12GB) because of the high resolution media captures. Factored representations [PvBM*06, CPZT12, KÖP13, Kur21] enable a very significant compression ratio for such data. With a similar representation, Deng et al. [DLW*22] introduce an inverse rendering technique for capturing translucent object appearance. The light transport is however limited for these models due to their use of an analytic BSSRDF approximation. An alternative approach is to represent parts of a volume path tracing. Kallweit et al. [KMM*17] assume no refractive interface and train a network to represent the multiple in-scattering during a volume path tracing with standard evaluation of single scattering. Rittig et al. [RSB*21] let a network represent subsurface scattering in a local 3D patch and assume diffuse incident illumination. This limits the use of the method to highly scattering materials and the influence of the direction of incidence is not a part of the model. TG et al. [TTJ*24] introduce a method to learn the full BSSRDF of a closed object with heterogeneous properties by training an object/material-specific neural network using a set of paths from volume rendering. While not restricted to distant lighting, this method requires many network evaluations to render a converged image, leading to non-interactive rendering performance.

Texture space techniques Like neural BTF [RGJW20, KMX*21], Zhang et al. [ZFT*21] work in texture space and assume a diffuse rendered base for which a network predicts the residual and Fan et al. [FWH*23] present a BTF compression technique using a universal MLP and two feature planes (for spatial and angular dependency) for fast compression, inference, and acquisition. We find it likely that these methods produce translucency effects similar to the neural BTF method we compare with [KMX*21]. Zheng et al. [ZZW*21] introduce a BRDF compression technique that represents continuous functions from sparsely measured datasets in a latent space using neural processes. While this method achieves high compression levels for general datasets, a BRDF-specific approach cannot replicate the effects of subsurface scattering.

Baatz et al. [BGP*22] include ray marching of a volumetric shell around an object to include mesoscale geometry like fur. As opposed to this, we include the entire heterogeneous volume and a refractive interface while avoiding the ray marching. However, we train a network per object whereas they train networks for different patches and apply these to an arbitrary surface.

3. Theory

For a translucent object, the radiance leaving a point of observation $\boldsymbol{x}_o$ in the direction $\vec{\omega}_o$ is given by the reflected radiance equation [Pre65, JMLH01]

$$L_r(\boldsymbol{x}_o,\vec{\omega}_o) = \int_A \int_{2\pi} S(\boldsymbol{x}_i,\vec{\omega}_i;\boldsymbol{x}_o,\vec{\omega}_o) L_i(\boldsymbol{x}_i,\vec{\omega}_i)\cos\theta_i \,\mathrm{d}\omega_i\,\mathrm{d}A\,, \quad (1)$$

where L_i is the incident radiance from the direction $\vec{\omega}_i$ at position $\boldsymbol{x}_i \in A$, so that the angle of incidence is θ_i, and S is the BSSRDF. We let A denote the surface area of the object, and we use arrow overline for unit length direction vectors (e.g. $\vec{\omega}_o$). We can evaluate this expression (1) if we have a known BSSRDF or by a random walk (volume path tracing) if we have an object with known surface and volume scattering properties [Pre65, PH00, FJM*20].

Assuming distant illumination, incident radiance is the same for positions in the object surface that are not in shadow. Letting V denote visibility, we have

$$L_i(\boldsymbol{x}_i,\vec{\omega}_i) = V(\boldsymbol{x}_i,\vec{\omega}_i)L_i(\vec{\omega}_i)\,, \quad (2)$$

Using a Dirac delta-function δ and a surface BRDF f_r, we separate out surface reflection to let this include local illumination:

$$S(\boldsymbol{x}_i,\vec{\omega}_i;\boldsymbol{x}_o,\vec{\omega}_o) = \delta(\boldsymbol{x}_o-\boldsymbol{x}_i)f_r(\boldsymbol{x}_o,\vec{\omega}_i,\vec{\omega}_o) + \bar{S}(\boldsymbol{x}_i,\vec{\omega}_i;\boldsymbol{x}_o,\vec{\omega}_o)\,. \quad (3)$$

We then write up a BSSRDF for subsurface scattering under distant illumination:

$$\hat{S}(\vec{\omega}_i;\boldsymbol{x}_o,\vec{\omega}_o) = \int_A \bar{S}(\boldsymbol{x}_i,\vec{\omega}_i;\boldsymbol{x}_o,\vec{\omega}_o)V(\boldsymbol{x}_i,\vec{\omega}_i)(\vec{\omega}_i\cdot\vec{n}_i)\,\mathrm{d}A\,, \quad (4)$$

where $\vec{n}_i$ is the surface normal at $\boldsymbol{x}_i$. Assuming A to be the surface of a closed object, light can be incident from all directions. We thus need to change our integration in Eq. 1 to be across the unit sphere (4π solid angles), and we note that when the cosine term is negative we always have zero visibility ($\vec{\omega}_i\cdot\vec{n}_i < 0 \Rightarrow V = 0$). The reflected radiance equation (1) then becomes

$$\boxed{L_r(\boldsymbol{x}_o,\vec{\omega}_o) = \int_{4\pi} L_i(\vec{\omega}_i)\,\hat{S}(\vec{\omega}_i;\boldsymbol{x}_o,\vec{\omega}_o)\,\mathrm{d}\omega_i\,.} \quad (5)$$

With an MLP representation of $\hat{S}$, we can use this equation for computing reflected radiance directly as a shader in a rendering system. The difference between S and $\hat{S}$ is illustrated in Fig. 2.

3.1. Neural Precomputed Subsurface Scattering

Let ψ denote the parameters of a multilayer perceptron (MLP) f_ψ. The purpose of the MLP is to represent the appearance of a translucent object. To this end, we optimize the parameters to account for a vast set of observations captured in rendered images illuminated by

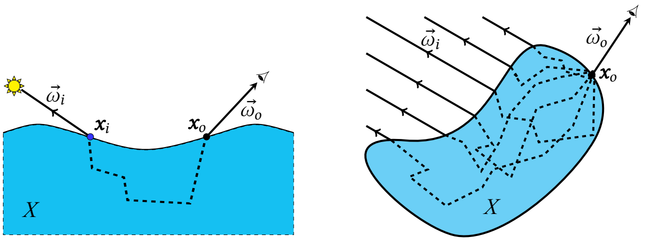


Figure 2: *The BSSRDF S (left) and the version for distant illumination $\hat{S}$ (right) that we use for our neural precomputed subsurface scattering. Both functions implicitly depend on the full object geometry X.*

directional lights of unit radiance $L_I = 1\,\frac{\mathrm{W}}{\mathrm{m}^2\,\mathrm{sr}}$. A directional light of direction $\vec{\omega}_e$ illuminates the object in each image. Letting $\boldsymbol{x}_{o,k}, \vec{\omega}_{o,k}$ denote the position and direction of observation for a ray through pixel k, we find parameters $\psi*$ for the MLP to represent the observation by

$$\psi* = \arg\min_{\psi} \mathbb{E}_k[(f_\psi(-\vec{\omega}_e;\boldsymbol{x}_{o,k},\vec{\omega}_{o,k}) - L_{r,k})^2], \qquad (6)$$

where $L_{r,k} \approx \hat{S}(-\vec{\omega}_e;\boldsymbol{x}_{o,k},\vec{\omega}_{o,k})L_I$ is the reflected radiance observed along the ray through pixel k. To make the network useful for rendering with arbitrary placement of the object in a scene, the position and direction of observation is given to the network in object space coordinates. The MLP $f_{\psi*}$ will serve as a representation of $\hat{S}$ in Eq. 5. External reflections at $\boldsymbol{x}_o$ are not included in the MLP. Refraction into and out of the object and internal reflections are all included in the MLP. We use a surface scattering model [WMLT07] for the reflection component f_r.

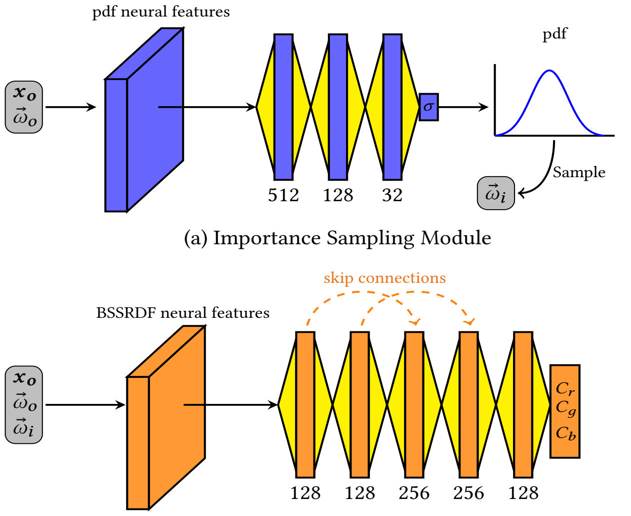
3.2. Importance Sampling

When using our model in a renderer, we evaluate Eq. 5 with our neural network $f_{\psi*}$ in place of $\hat{S}$. For Monte Carlo integration of Eq. 5, we sample directions of incidence $\vec{\omega}_i$ according to a probability density function $\mathrm{pdf}(\vec{\omega}_i)$. The Nth estimator of the integral is then

$$L_r(\boldsymbol{x}_o,\vec{\omega}_o) \approx \frac{1}{N}\sum_{j=1}^{N}\frac{L_i(\vec{\omega}_{i,j})\,f_{\psi*}(\vec{\omega}_{i,j};\boldsymbol{x}_o,\vec{\omega}_o)}{\mathrm{pdf}(\vec{\omega}_{i,j})}, \qquad (7)$$

and importance sampling is performed by choosing the pdf. One option is importance sampling of the L_i term by importance sampling an environment map [PJH23]. This works well with our technique, but it has the drawback of sampling directions that are not necessarily important for the object appearance in a given point of observation $\boldsymbol{x}_o$. As an alternative, we would like an importance sampling technique with a pdf roughly proportional to the value of $\hat{S}$ for different directions of incidence across the unit sphere.

As in existing work on importance sampling for neural representations of spatially varying BRDFs [FWH*22], we sample the direction of incidence $\vec{\omega}_i$ using a Gaussian lobe. However, in our case, the light incident from any direction on the unit sphere can be important. Due to Fresnel transmittance, subsurface scattering is usually more important when light is incident closer to the surface normal $\vec{n}_o$ at the point of observation. For our pdf, we thus use a zero mean, normalized 2D Gaussian kernel with standard deviation σ in the azimuthal equidistant projection around $\vec{n}_o$ (this corresponds to a normalized spherical Gaussian with its axis along $\vec{n}_o$). The σ value becomes large when backlighting is important. We use the Box-Muller transform [HT07] to sample direction vectors according to the σ retrieved from the network.



(a) Importance Sampling Module

(b) Appearance Specification Module

Figure 3: *Our network architecture consists of two parts: an importance sampling module and an appearance specification module.*

4. Implementation

Our subsurface scattering model is an MLP representation of $\hat{S}$ (Eq. 4) that takes as input $\vec{\omega}_i, \vec{\omega}_o, \boldsymbol{x}_o$ (7-vector input: 2 dimensions for each direction in spherical coordinates and 3 for the position in Cartesian coordinates) and returns relative outgoing radiance (3-vector if RGB) for unit radiance incident at every illuminated object surface position (see Fig. 2). The result is scaled to the radiance of a sampled distant light to get the contribution in a Monte Carlo renderer. Our neural network only captures the subsurface scattering. Surface scattering effects [WMLT07] are computed separately.

4.1. Network Architecture

Our architecture consists of two major components: an importance sampling module and an appearance specification module, see Fig. 3. Due to the known inability of MLPs to learn high frequency functions in low dimensional domains [TSM*20] and due to bias towards low-frequency functions, we cast our problem into a higher dimensional space before passing it to the MLP. As seen in Fig. 3, both modules therefore have two components: feature extraction and neural network. The feature extraction is similar in both the modules. We use Fourier feature maps defined by

$$\boldsymbol{F}(\boldsymbol{v}) = [\cos(2\pi\boldsymbol{G}\boldsymbol{v}), \sin(2\pi\boldsymbol{G}\boldsymbol{v})]^\top, \qquad (8)$$

where $\boldsymbol{v}$ is an input vector of size n (5 for importance sampling, 7 for appearance specification) and $\boldsymbol{G}$ is a static matrix of size $256 \times n$ randomly sampled at initialization using a 2D Gaussian distribution with a mean sufficiently large to get a positive matrix (we use $\mu_{\boldsymbol{G}} = 18$) and a standard deviation of $\sigma_{\boldsymbol{G}} = 1$. Since $\boldsymbol{G}$ is a randomly sampled isotropic distribution, it performs frequency modulation on the feature map. Furthermore, the standard deviation ($\sigma_{\boldsymbol{G}}$) is a significant factor in extracting the features, however $\sigma_{\boldsymbol{G}} = 1$ worked well in all our tests, presumably because it sustains the bell curve of the distribution.

The neural network in the importance sampling module is a 3-layers MLP with 512, 128, and 32 neurons, respectively, and ReLU (Rectified Linear Unit) activations between each layer but none after the last layer. This network should predict a standard deviation (σ) for the importance sampling Gaussian lobe making it as similar as possible to the output distribution of the appearance specification module (for the same $\boldsymbol{x}_o, \vec{\omega}_o$). Since the aim is to reduce the difference of two distributions, we use Kullback-Leibler divergence [KL51] loss for fitting our distribution prediction.

The appearance specification module is an MLP with five to elleven layers, 128 neurons in each layer, and skip connections between alternate layers. In Sec. 5, we demonstrate how the skip connections lead to better quality with fewer layers, enabling a small memory footprint compared with previous representations of translucent objects. We use ReLU as activation after each layer, except for the last layer after which we use no activation, and we use L2 as the loss function between our prediction and ground truth.

4.2. Training and Dataset

Our model is trained on data produced by path tracing a scene consisting of the object of interest and a directional light source with unit radiance. The direction of the light source is randomly changed and the camera is orbiting the center of the object geometry. We used 10000 spp and a path-length limit of 5000 and filter the data to only include rays that hit the object.

We use an online learning method to train the appearance specification module. With this method, the renderer and the neural network run in parallel, requiring no overhead for training. The task of the renderer is to compute data at all times. When new data becomes available, the renderer pushes it into a buffer used by the network for learning. We use the running L2 loss and the L2 loss on a small test set (produced prior to training) to check the network convergence. This approach takes 24-48 hours, which is equal to the time needed for data generation, with no additional time for training. Another approach to do the same is an offline method, where we collect and store the necessary data on disk and train the model independently. This can be done with a general purpose renderer like Mitsuba [NDVZJ19] or PBRT [PJH23]. For quantitative analysis, we have tested an offline learning method by collecting data from a renderer and training the model as a secondary step. We used 5% of the data as a test set and another 5% for validation. We found that the collected data $\hat{S}(\vec{\omega}_i; \boldsymbol{x}_o, \vec{\omega}_o)$ from 835 images (4,659,144 data points), where each image had a distinct $\vec{\omega}_i$, were enough to have the network represent a heterogeneous medium. The images do not directly correspond to the amount of data acquired. Every view might have a different number of hits on the geometry. Furthermore, training data was generated in low resolution and lower sample counts to speed up the data acquisition. This method requires 3-4 hours of training after data generation per object. An additional 25 minutes are required for either method if one would like to also convert to spherical harmonic PRT (Sec. 4.3).

The importance sampling module is trained on inference data from the appearance specification module. Thus, no new data are generated to train this module. We randomly select $\boldsymbol{x}_o$ and $\vec{\omega}_o$ and use a batch of 32×64 uniformly distributed spherical coordinates defining $\vec{\omega}_i$. The resulting data are collected into bins that form our ground truth distribution. This distribution is expected to be approximately Gaussian (discussed in Sec. 3.2). In this way, we obtain a network predicting the standard deviation (σ) for the Gaussian lobe used for importance sampling.

4.3. Converting to Spherical Harmonic PRT

If one is willing to trade accuracy for rendering efficiency, our neural precomputed subsurface scattering representation is easily converted to conventional spherical harmonic PRT. Traditional PRT techniques rely on efficient sampling of spherical functions such as environment illumination and diffuse irradiance to construct signals in the form of basis vectors and coefficients. Due to low dimensionality requirements, these PRT methods have not been very effective in addressing higher dimensional problems, such as a BSSRDF with full directionality and large variation in scattering properties.

The lighting coefficients are computed in the standard way,

$$b_l^m = \int_{4\pi} L_i(\vec{\omega}_i) Y_l^m(\vec{\omega}_i)\, \mathrm{d}\omega_i\,, \tag{9}$$

where L_i is incident radiance from the environment map and Y_l^m are the (real) spherical harmonic basis functions. The spherical harmonic coefficients of the light transport are obtained from our neural representation,

$$a_l^m(\boldsymbol{x}_o, \vec{\omega}_o) = \int_{4\pi} \hat{S}(\vec{\omega}_i; \boldsymbol{x}_o, \vec{\omega}_o) Y_l^m(\vec{\omega}_i)\, \mathrm{d}\omega_i\,, \tag{10}$$

where the subsurface scattering function $\hat{S}$ is integrated with Y_l^m, and we use the MLP model f_{ψ^*} for $\hat{S}$. We use degree 2 spherical harmonics $l \leq 2$ with nine a_l^m and b_l^m coefficients to represent $\hat{S}$. Our neural representation conveniently provides inexpensive low-variance estimates of the integral for the coefficients a_l^m.

The computation of a_l^m occurs at each vertex of the object surface for every direction $\vec{\omega}_o$ on the hemisphere. In practice, we distribute $\vec{\omega}_o$ into equal solid angles on the hemisphere for every vertex and Monte Carlo sample $\vec{\omega}_i$ over the sphere to construct the coefficients a_l^m for each solid angle. At render time, we interpolate the coefficients for the observed $\vec{\omega}_o$ and compute the result using

$$L_r(\boldsymbol{x}_o, \vec{\omega}_o) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} a_l^m(\boldsymbol{x}_o, \vec{\omega}_o) b_l^m\,. \tag{11}$$

An SH representation has inherent limitations in accurately representing high-frequency functions. To surpass the limitations of spherical harmonics in capturing intricate details with precision, it is possible to explore alternative representations like wavelets [NRH03, WTL05, RXL*23]. Wavelets are well-known for their superior ability to represent high-frequency functions, but capturing full view- and light-dependent subsurface scattering effects remains a significant challenge.

Table 1: *Performance and accuracy analysis for our method. The arrows denote whether smaller (↓) or larger (↑) values are better.*

Experiment Name	Model	SPP	Render Time (sec.)↓	Storage (MB)↓	MSE↓	SSIM↑	ꟻLIP mean↓
Grape	Path Tracing	10000	1846.44				
	Ours(Net)	256	2.038	1.31	$1.6 \cdot 10^{-4}$	0.9820	0.051
	Ours(SH)	1	$3.2 \cdot 10^{-4}$	21.1	$2.8 \cdot 10^{-3}$	0.9816	0.082
Dragon	Path Tracing	10000	2412.56				
	Ours(Net)	256	1.813	1.31	$1.4 \cdot 10^{-5}$	0.9985	0.009
	Ours(SH)	1	$1.2 \cdot 10^{-3}$	779.6	$6.2 \cdot 10^{-4}$	0.9866	0.045
Paperweight	Path Tracing	10000	2503.83				
	Ours(Net)	256	3.950	3.77	$1.1 \cdot 10^{-3}$	0.8372	0.079
	Ours(SH)	1	$3.6 \cdot 10^{-4}$	46.1	$3.4 \cdot 10^{-3}$	0.9206	0.115
Drink with ice	Path Tracing	10000	1496.24				
	Ours(Net)	256	3.231	1.31	$3.0 \cdot 10^{-4}$	0.9629	0.043
	Ours(SH)	1	$2.8 \cdot 10^{-4}$	19.2	$1.4 \cdot 10^{-3}$	0.9620	0.070
Lucy	Path Tracing	10000	1198.36				
	Ours(Net)	256	2.189	0.65	$4.7 \cdot 10^{-4}$	0.9906	0.035
	Ours(SH)	1	$3.7 \cdot 10^{-4}$	64.2	$1.1 \cdot 10^{-3}$	0.9772	0.382

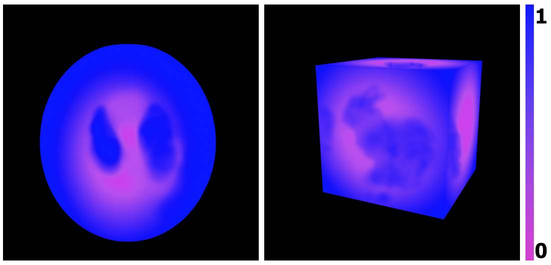


Figure 4: *Visualizations of the standard deviation σ predicted by our importance sampling module. We generate σ once per pixel and then sample $\vec{\omega}_i$ for the appearance specification module using the generated distribution.*

4.4. Framework and Rendering

During network training, we employ our custom-built GPU rendering framework that leverages NVIDIA OptiX [PBD*10] for efficient data collection. The neural network is trained on a C++ distribution of PyTorch [PGM*19] (popularly known as libTorch). However, we extract weights from the PyTorch model and use them in our own implementation when computing network inference. Our framework can be integrated into Monte Carlo renderers easily if we do no batch queries at inference time and evaluate each shading point independently. We used an NVIDIA RTX 4090 in all tests.

Rendering is done in a shader where our importance sampling module infers the standard deviation (σ) of the expected Gaussian pdf, see examples in Fig. 4. A sample direction $\vec{\omega}_{i,j}$ is drawn from the predicted Gaussian pdf. Our appearance specification module is inferred for a unit radiance incident light based on the inputs (i.e. $\boldsymbol{x}_o, \vec{\omega}_o, \vec{\omega}_i$), which in turn is scaled for the actual contribution from the distant illumination. Finally, surface scattering is added.

To optimize the performance, we do batch inference of the network inside the master/ray-generation program yielding a 15-20% performance boost as compared to individual inference inside the shader program.

For our spherical harmonic PRT simulation, we precompute the coefficients by Monte Carlo integration, sampling $\vec{\omega}_i$ for an even 16×64 distribution of $\vec{\omega}_o$ directions at each vertex in object space. The combinations of directions in each vertex are used for inference of our appearance specification module to obtain the a_l^m coefficients. A simpler Monte Carlo integration of incident illumination is used to obtain b_l^m. When rendering, the desired output is a simple dot product of the resulting coefficients (Eq. 11).

5. Results

To test whether our method can represent different optical thicknesses, we first trained our model multiple times using the same heterogeneous object with different levels of translucency. The results in Fig. 5 demonstrate that our model represents the directional dependence of incoming radiance with good accuracy, however the accuracy decreases with larger mean free path. We attribute this to higher variance in the training data for media with low scattering. The lower the scattering, the harder it is for the path tracing to find a path with a contribution from the source. Path guiding techniques [VHH*19] could be used to improve on this issue and could likely lower the training time in general. A test with a single directional light illuminating an object with non-trivial surface geometry is in Fig. 6. Note that our model captures the high frequency self-shadowing of the translucent object well. A performance and accuracy analysis is provided for five different objects in Table 1. The objects are shown in Fig. 8. Their shapes and materials are:

- *Grape* consists of a scattering volume behind a refractive interface with spatially varying surface roughness defined by a texture map. The volume contains two seeds that are diffuse with reflectance defined by an image texture.

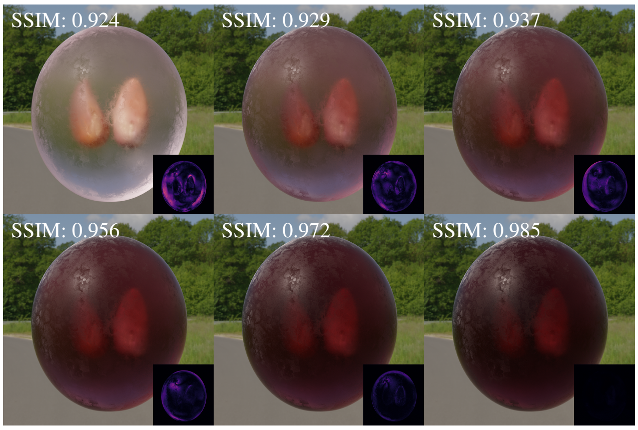


Figure 5: *Our method applied to objects with different levels of translucency (from top left to bottom right, the mean free path is longer by a factor 200, 100, 20, 10, 2, 1). The inserts at the bottom right show the ꟻLIP mean between the images rendered using path tracing and our neural representation. SSIM indices (top) demonstrate high accuracy compared to the ground truth.*

- *Dragon* is the XYZ RGB dragon from the 3D scanning repository of the Stanford Computer Graphics Laboratory. We use a noise function to define the optical properties inside the model.
- *Paperweight* is a cube with a scattering volume. It contains the Stanford bunny (Stanford CG Lab) with solid texture variation of the optical properties of marble from Jensen et al. [JMLH01].
- *Drink with ice* is a scattering medium in a drinking glass. The model includes two ice cubes inside the liquid.
- *Lucy* is another model from the Stanford CG Lab which we augmented with solid texture variation of ketchup and marble materials. The optical properties of these translucent materials are from Jensen et al. [JMLH01].

Three of the five objects are rendered side-by-side in the same environment in Fig. 1. We employ three objective full image quality metrics: mean-squared error (MSE), structural similarity (SSIM) index [WBSS04], and the mean value of ꟻLIP [ANAM*20]. Our network achieves high fidelity (ꟻLIP below 0.1), enabling us to represent the complex appearance of the objects while efficiently storing the data. Our appearance specification module can run at more than 60 samples per pixel per second, meaning that our method supports real-time rendering with one sample per pixel (SPP) and progressive updates. Additionally, conversion to spherical harmonic PRT achieves real-time rendering frame rates (less than 2 ms per frame) at a higher storage cost.

Ablation study. We conducted an ablation test by evaluating the performance of various network sizes in comparison to a 7-layered network, see Fig. 7. A 7-layered network was sufficient for representing the complex heterogeneous translucent objects in our tests to a high fidelity (ꟻLIP below 0.1). It is worth noting that we incorporate skip connections between alternate layers as our tests demonstrate that this architectural design choice leads to better accuracy with fewer layers and thus better compression and improved rendering performance.

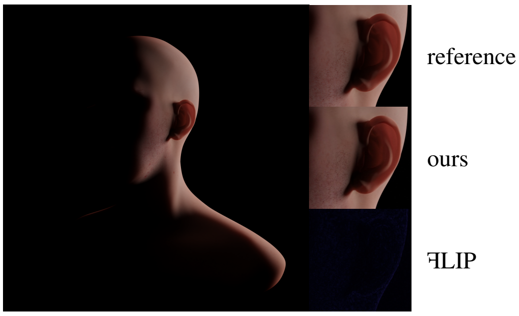


Figure 6: *Our method accurately represents self-shadowing in translucent objects with non-trivial geometry, such as the ear casting a shadow under backlighting while allowing light to bleed through. This example shows the neural BSSRDF works equally well for a single directional light as for the environment light in other examples. The ꟻLIP error of the close-up on the ear shows high accuracy as compared with the path traced reference image.*

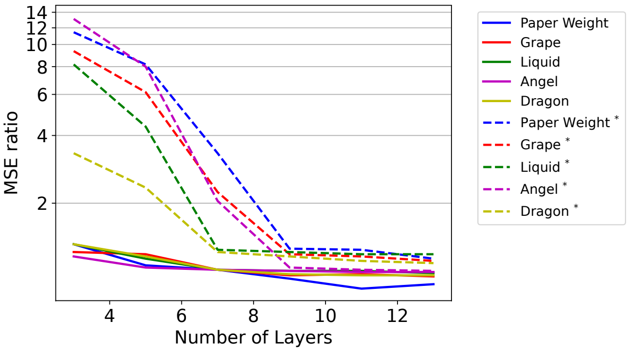


Figure 7: *The influence of varying the number of layers in our appearance specification module on the resulting MSE taken relative to a 7-layers MLP with skip connections. The asterisk $*$ marks the same results but without skip connections. This clearly demonstrates the importance of the skip connections and illustrates the trade-off in selecting the number of layers (for each object we chose a number of layers based on where the curve bends).*

Low storage. During the design of our neural representation, we considered storage efficiency important. Comparing with other techniques for storing heterogeneous scattering volumes, such as that of Kurt et al. [KÖP13], our model achieves a more compact representation. However, since we did not have access to the same dataset, a direct quantitative comparison was not possible. Table 1 reports the storage efficiency of our model. Our storage requirements are lower than the compact factored representations [PvBM*06, KÖP13] (for which 4.2 to 39 MB were spent to represent measured heterogeneous translucent materials).

Comparison with other methods. Table 2 provides a quantitative analysis and Fig. 9 enables a qualitative assessment of the accuracy and performance of our method as compared with other object and material appearance representation methods. We conducted a comparison with NeRFactor, NeuMIP, and textured BSSRDF. In our comparison, we let NeRFactor [ZSD*21] represent relightable

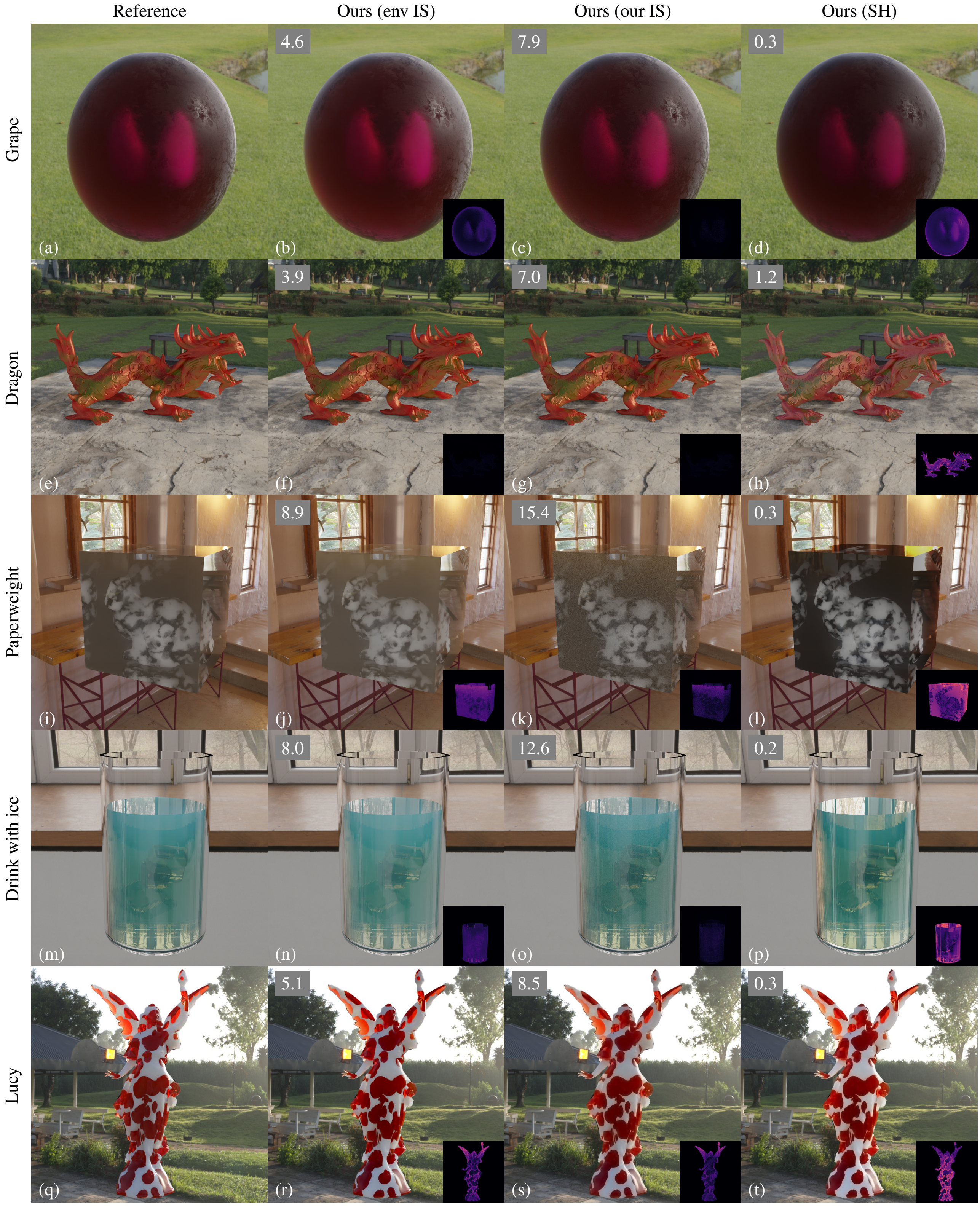


Figure 8: *The ability of different versions of our model to represent path traced reference images. We show traditional importance sampling of the environment map [PJH23] (env IS) compared against our importance sampling technique described in Sec. 3.2 (our IS) and the SH representation of our model for real-time rendering described in Sec. 4.3 (SH). Each image with importance sampling (IS) is rendered at 256 samples per pixel (SPP). The bottom right corners of the images show the ꟻlip error metric and the top left corner shows the average render time per 1 SPP frame in milliseconds (ms).*

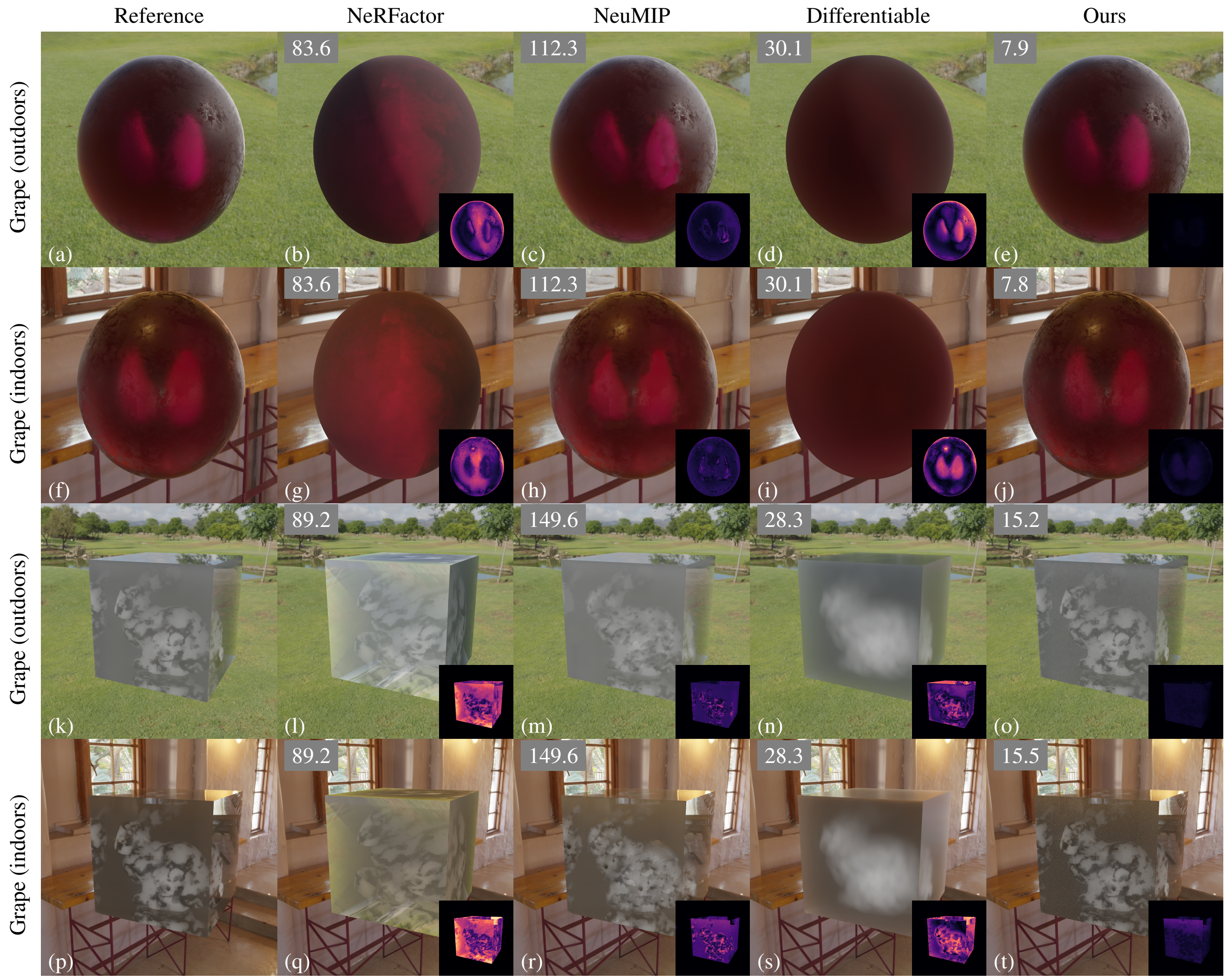


Figure 9: *Comparison with other neural methods for object or material appearance capture. We compare with NeRFactor [ZSD*21], which is a spatially varying BRDF approximation that enables change of the lighting environment, NeuMIP [KMX*21], a texture-based material appearance representation method, and Differentiable [DLW*22], a differentiable rendering method for fitting an analytic BSSRDF to capture translucent object appearance. The bottom right corners of the images show the ꟻlip error metric and the top left corner shows the average render time per 1 SPP frame in milliseconds (ms).*

NeRF models that infer BRDF parameters using a handful of images. NeuMIP [KMX*21] represents patch-based BTF techniques. The NeuMIP architecture is not well-suited for representing subsurface scattering due to its nature as a texture function. To enable a fair comparison with NeuMIP, we render textures in UV space to train the networks. However, this approach results in certain components of the NeuMIP architecture becoming obsolete, such as the neural offset module, as they no longer serve a purpose in our context. To compare with conventional textured surface variation of the parameters in the analytic standard dipole BSSRDF model [JMLH01], we use a method that estimates the textured parameters needed to represent the appearance of a given object using differentiable rendering [DLW*22].

NeRFactor learns a BRDF scene and thus assumes absence of subsurface scattering. Consequently, the resulting appearance resembles that of a flat texture on the model. This limitation leads to a lack of perceptible depth, as is evident in the second column of Fig. 9. In contrast, NeuMIP relies on a texture pyramid to efficiently store material properties. However, this approach introduces limitations in terms of the complexity of materials that can be effectively learned. Additionally, the trilinear interpolation used for the texture pyramid introduces blur, particularly in high-frequency details. This can be seen in the third column of our tests in Fig. 9. Our method addresses this issue by incorporating Fourier feature extraction, allowing the network to learn and capture high-frequency details more effectively. The analytic BSSRDF approach is challenged by the fact that these models were derived for a homogeneous translucent planar half-space.

Table 2: *Image quality metrics with path traced reference images for NeRFactor [ZSD*21], NeuMIP [KMX*21], analytic BSSRDF fitted by differentiable rendering (Diff.) [DLW*22], and the network version of our method. Image resolution was* 1920^2 *(MSE, SSIM1, ꟻLIP) with 256 SPP (10k SPP for the reference). The SSIM metric prefers blur to noise. If we downsample the images to* 480^2*, some of the SSIM values change significantly (SSIM2).*

	Method	MSE↓	SSIM1↑	SSIM2↑	ꟻLIP↓
Grape (out)	NeRFactor	$3.34 \cdot 10^{-3}$	$8.89 \cdot 10^{-1}$	$8.97 \cdot 10^{-1}$	$1.32 \cdot 10^{-1}$
	NeuMip	$1.82 \cdot 10^{-4}$	$\mathbf{9.91 \cdot 10^{-1}}$	$9.94 \cdot 10^{-1}$	$5.12 \cdot 10^{-2}$
	Diff.	$3.86 \cdot 10^{-3}$	$9.16 \cdot 10^{-1}$	$9.19 \cdot 10^{-1}$	$1.15 \cdot 10^{-1}$
	Ours(Net)	$\mathbf{9.28 \cdot 10^{-5}}$	$9.82 \cdot 10^{-1}$	$\mathbf{9.99 \cdot 10^{-1}}$	$\mathbf{1.74 \cdot 10^{-2}}$
Grape (in)	NeRFactor	$2.84 \cdot 10^{-3}$	$9.41 \cdot 10^{-1}$	$9.40 \cdot 10^{-1}$	$1.37 \cdot 10^{-1}$
	NeuMip	$9.81 \cdot 10^{-4}$	$\mathbf{9.85 \cdot 10^{-1}}$	$9.84 \cdot 10^{-1}$	$1.60 \cdot 10^{-1}$
	Diff.	$1.89 \cdot 10^{-3}$	$9.48 \cdot 10^{-1}$	$9.46 \cdot 10^{-1}$	$1.06 \cdot 10^{-1}$
	Ours(Net)	$\mathbf{3.11 \cdot 10^{-4}}$	$9.67 \cdot 10^{-1}$	$\mathbf{9.98 \cdot 10^{-1}}$	$\mathbf{3.01 \cdot 10^{-2}}$
Paperw. (out)	NeRFactor	$9.31 \cdot 10^{-3}$	$8.87 \cdot 10^{-1}$	$8.36 \cdot 10^{-1}$	$1.63 \cdot 10^{-1}$
	NeuMip	$7.40 \cdot 10^{-4}$	$\mathbf{9.86 \cdot 10^{-1}}$	$9.64 \cdot 10^{-1}$	$6.08 \cdot 10^{-2}$
	Diff.	$2.58 \cdot 10^{-3}$	$9.50 \cdot 10^{-1}$	$9.09 \cdot 10^{-1}$	$8.84 \cdot 10^{-2}$
	Ours(Net)	$\mathbf{6.91 \cdot 10^{-4}}$	$8.08 \cdot 10^{-1}$	$\mathbf{9.81 \cdot 10^{-1}}$	$\mathbf{2.79 \cdot 10^{-2}}$
Paperw. (in)	NeRFactor	$1.01 \cdot 10^{-2}$	$9.13 \cdot 10^{-1}$	$9.40 \cdot 10^{-1}$	$1.70 \cdot 10^{-1}$
	NeuMip	$1.95 \cdot 10^{-3}$	$\mathbf{9.75 \cdot 10^{-1}}$	$9.84 \cdot 10^{-1}$	$8.90 \cdot 10^{-2}$
	Diff.	$6.20 \cdot 10^{-3}$	$9.53 \cdot 10^{-1}$	$9.46 \cdot 10^{-1}$	$1.21 \cdot 10^{-1}$
	Ours(Net)	$\mathbf{1.15 \cdot 10^{-3}}$	$8.37 \cdot 10^{-1}$	$\mathbf{9.98 \cdot 10^{-1}}$	$\mathbf{4.91 \cdot 10^{-2}}$

In Table 2, the SSIM value for our network representation was lower than for NeuMIP. This is due to SSIM prefering blur to noise. When we importance sample the BSSRDF instead of the incident illumination, some stochastic noise is still present after 256 samples per pixel. SSIM picks up on this stochastic noise (SSIM1 in Table 2). If we downsample the image (SSIM2 in Table 2) or use importance sampling of the environment and full resolution (second column of Fig. 8), we get a higher SSIM than the other methods. Simply allowing our method that importance samples the BSSRDF more samples per pixel is another way to get higher SSIM in the full resolution image.

Denoised path tracing. As volume path tracing is expensive in terms of render time, the training cost of our method is justified whenever we need an ability to interactively change view or lighting of a heterogeneous translucent object. To illustrate the differences between our approach once trained and volume path tracing, Fig. 10 provides an equal render time comparison of the two methods when renderings are performed for 1 second. Since the unconverged images resulting from path tracing are very noisy, we also shorten the maximum path length to 500 and use the OptiX denoiser to let the images reach a reasonable noise level. The path traced images appear darker when unconverged due to noisy high intensity pixels (fireflies) being clamped when the image is displayed.

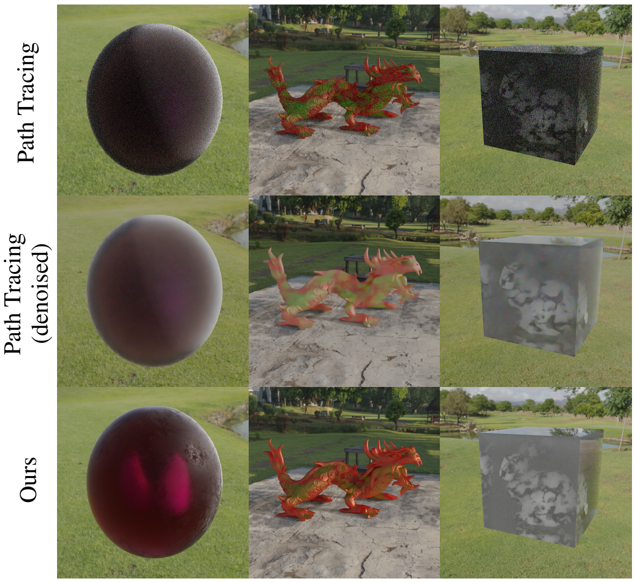


Figure 10: *Equal render time results (1 second) for our method and volume path tracing. Comparing these with the converged path traced image, the MSE of our method is two orders of magnitude smaller and ꟻlip is one order of magnitude smaller.*

Quality of representations. Fig. 8 provides a comparison of different uses of our appearance representation with path traced reference images. An advantage of our approach is that one can freely select the sampling technique. To demonstrate the effect of different techniques, we compare use of our appearance specification module with environment importance sampling [PJH23], use of our own importance sampling module, and use after conversion of our model to an SH representation. The scenes are the same as in the quantitative analysis in Table 1.

When using environment importance sampling, we more often sample similar sources in the observed points. This reduces stochastic noise and the error becomes less obvious as in denoised images. More samples are however required to accurately account for the darker and more infrequently sampled directions that might be important for the appearance of the translucent object. Our importance sampling module puts more focus on our appearance specification module, which can oftentimes result in reduced error at equal sample counts. Instead of using multiple importance sampling, we render the images at 256 samples per pixel with each of the two techniques to clearly show their respective effects. Depending on the scene to be rendered, use of multiple importance sampling may be advantageous, and a network can be used to find multiple importance sampling weights [FWH*22]. The conversion of our model to an SH representation demonstrates the accuracy-efficiency trade-off. While the ꟻLIP error rates are higher in the SH representation compared to our neural representation, the objects do maintain some visual fidelity. For a visual comparison with a dynamic camera, we refer the reader to the supplementary video.

6. Discussion

Importance sampling Our importance sampling works in general but can in some cases be a crude approximation of the light distri-

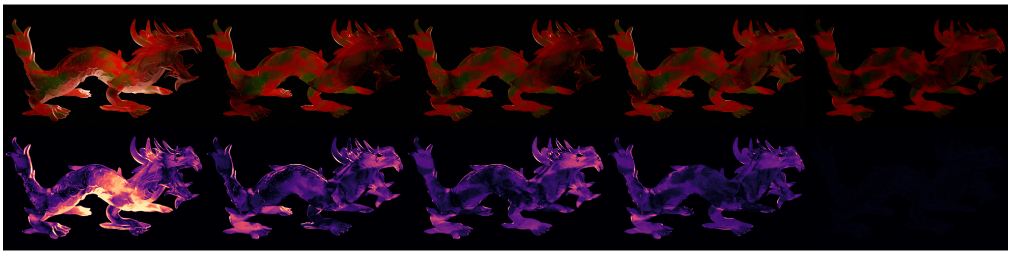


Figure 11: *To test our method in the case of local lighting, we render the dragon with a point light behind the model (top row). From left to right, we gradually move the point light away from the model and use a directional light for the rightmost image. The ꟻlip error with volume path tracing as the reference are in the bottom row.*

bution function. The efficiency of the method is therefore different for different scenarios. In some cases, like when rendering with one directional light (as in Fig. 6), light source sampling is of course better and only one sample is needed. In a more diffuse lighting environment, the simple Gaussian importance sampling with spatially varying σ tends to work well. In our implementation, we use a discretization of the unit sphere into equal solid angles, which is common for various types of BRDFs. However, this approach may not be suitable for subsurface scattering due to potential non-representativeness of the distribution function. One such outlier in a Gaussian distribution could be that certain angles around the edges in the object geometry are more important than the nearby angles. As the Gaussian distribution is a monotonically decreasing function, it may not adequately represent such sharp changes. A sum of Gaussians [XWH*23] might be preferable.

Local lighting. Since we assume distant lighting, our method gradually introduces larger error as a light becomes more local. In Fig. 11, the dragon is illuminated by a point light placed behind the model. We use the direction from the center of the object bounding box to the position of the light to render the object with our model. As expected, the error in comparison with a volume path traced reference image is larger the closer the point light is to the object. When using our method, we adjusted the radiance of the directional light representing the point to best match the radiance on the surface due to the point. This experiment clearly shows the limitation of our model due to our assumption of distant lighting. However, it should be noted that the point light starts behaving like a directional light as it moves farther away from the object.

One extension of the model could be to use point lights as well as directional lights when training. However, we did not obtain good results with this approach. An alternative approach would be to learn the full BSSRDF of a heterogeneous translucent object. Such a method however requires more network evaluations to reach a converged image because each evaluation only considers light incident from a limited number of points across the object surface [TTJ*24]. As a consequence, the rendering time is significantly higher (not interactive).

7. Conclusion

We have presented a method for representing the appearance of a heterogeneous translucent object. Our approach leverages an MLP for compact storage, reproduces arbitrary views, and is not limited to low-frequency relighting. In fact, we interactively render noise-free images of a translucent object illuminated by a directional source in near-path-tracing quality. An object placed in a more complicated lighting environment requires sampling of light directions. This introduces some sampling noise that can either be reduced by importance sampling the environment or by importance sampling the object appearance using our importance sampling network which is trained using our network representing the object appearance. In addition, when our appearance MLP is available, it can be used to easily compute conventional PRT spherical harmonics coefficients, trading compactness and accuracy for speed, which we find a compelling feature of our representation. In conclusion, we provide a method that uses precomputed lightweight neural networks for interactive rendering in near-path-tracing quality of heterogeneous scattering media with a refractive interface under distant lighting.

Acknowledgements This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 956585 (PRIME). This work was supported in part by NSF grant 2212085. We also acknowledge gifts from Adobe, Google, Qualcomm and Rembrand, and the Ronald L. Graham Chair.

References

[ANAM*20] Andersson P., Nilsson J., Akenine-Möller T., Oskarsson M., Åström K., Fairchild M. D.: ꟻLIP: a difference evaluator for alternating images. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 3*, 2 (2020), 15:1–15:23. doi:10.1145/3406183. 7

[AWB11] Arbree A., Walter B., Bala K.: Heterogeneous subsurface scattering using the finite element method. *IEEE Transactions on Visualization and Computer Graphics 17*, 7 (2011), 956–969. doi:10.1109/TVCG.2010.117. 2

[BBJ*21] Boss M., Braun R., Jampani V., Barron J. T., Liu C., Lensch H.: NeRD: Neural reflectance decomposition from image collections. In *International Conference on Computer Vision (ICCV)* (2021), IEEE, pp. 12684–12694. doi:10.1109/ICCV48922.2021.01245. 2

[BDBS22] Belcour L., Deliot T., Barbier W., Soler C.: A data-driven paradigm for precomputed radiance transfer. *Proceedings of the ACM on Computer Graphics and Interactive Techniques 5*, 3 (July 2022), 26:1–26:15. doi:10.1145/3543864. 2

[BGP*22] Baatz H., Granskog J., Papas M., Rousselle F., Novák J.: NeRF-Tex: neural reflectance field textures. *Computer Graphics Forum 41*, 6 (2022), 287–301. doi:10.1111/cgf.14449. 3

[BMF*22] Bemana M., Myszkowski K., Frisvad J. R., Seidel H.-P., Ritschel T.: Eikonal fields for refractive novel-view synthesis. In *SIGGRAPH 2022 Conference Proceedings* (August 2022), ACM, pp. 39:1–39:9. doi:10.1145/3528233.3530706. 3

[CPZT12] Chen G., Peers P., Zhang J., Tong X.: Real-time rendering of deformable heterogeneous translucent objects using multiresolution splatting. *The Visual Computer 28* (2012), 701–711. doi:10.1007/s00371-012-0704-1. 2, 3

[DLW*22] Deng X., Luan F., Walter B., Bala K., Marschner S.: Reconstructing translucent objects using differentiable rendering. In *SIGGRAPH 2022 Conference Proceedings* (2022), ACM, pp. 38:1–38:10. doi:10.1145/3528233.3530714. 2, 3, 9, 10

[DWWH20] Deng H., Wang B., Wang R., Holzschuch N.: A practical path guiding method for participating media. *Computational Visual Media 6* (2020), 37–51. doi:10.1007/s41095-020-0160-1. 2

[FD17] Frederickx R., Dutré P.: A forward scattering dipole model from a functional integral approximation. *ACM Transactions on Graphics 36*, 4 (2017), 109:1–109:13. doi:10.1145/3072959.3073681. 2

[FHK14] Frisvad J. R., Hachisuka T., Kjeldsen T. K.: Directional dipole model for subsurface scattering. *ACM Transactions on Graphics 34*, 1 (2014), 5:1–5:12. doi:10.1145/2682629. 2

[FJM*20] Frisvad J. R., Jensen S. A., Madsen J. S., Correia A., Yang L., Gregersen S. K. S., Meuret Y., Hansen P.-E.: Survey of models for acquiring the optical properties of translucent materials. *Computer Graphics Forum 39*, 2 (2020), 729–755. doi:10.1111/cgf.14023. 3

[FWH*22] Fan J., Wang B., Hasan M., Yang J., Yan L.-Q.: Neural layered BRDFs. In *SIGGRAPH 2022 Conference Proceedings* (2022), ACM, pp. 4:1–4:8. doi:10.1145/3528233.3530732. 2, 4, 10

[FWH*23] Fan J., Wang B., Hašan M., Yang J., Yan L.-Q.: Neural biplane representation for BTF rendering and acquisition. In *SIGGRAPH 2023 Conference Proceedings* (2023), ACM, pp. 81:1–81:11. doi:10.1145/3588432.3591505. 3

[GCD*20] Gao D., Chen G., Dong Y., Peers P., Xu K., Tong X.: Deferred neural lighting: free-viewpoint relighting from unstructured photographs. *ACM Transactions on Graphics 39*, 6 (2020), 258:1–258:15. doi:10.1145/3414685.3417767. 2

[GLL*04] Goesele M., Lensch H. P. A., Lang J., Fuchs C., Seidel H.-P.: DISCO: acquisition of translucent objects. *ACM Transactions on Graphics 23*, 3 (2004), 835–844. doi:10.1145/1015706.1015807. 3

[HT07] Howes L., Thomas D.: Efficient random number generation and application using CUDA. In *GPU Gems 3*. NVIDIA/Pearson, 2007, pp. 805–830. 4

[JMLH01] Jensen H. W., Marschner S. R., Levoy M., Hanrahan P.: A practical model for subsurface light transport. In *SIGGRAPH 2001* (2001), ACM, pp. 511–518. doi:10.1145/383259.383319. 2, 3, 7, 9

[KL51] Kullback S., Leibler R. A.: On information and sufficiency. *The Annals of Mathematical Statistics 22*, 1 (1951), 79–86. URL: https://www.jstor.org/stable/2236703. 5

[KMM*17] Kallweit S., Müller T., Mcwilliams B., Gross M., Novák J.: Deep scattering: rendering atmospheric clouds with radiance-predicting neural networks. *ACM Transactions on Graphics 36*, 6 (2017), 231:1–231:11. doi:10.1145/3130800.3130880. 3

[KMX*21] Kuznetsov A., Mullia K., Xu Z., Hašan M., Ramamoorthi R.: NeuMIP: Multi-resolution neural materials. *ACM Transactions on Graphics 40*, 4 (2021), 175:1–175:13. doi:10.1145/3450626.3459795. 2, 3, 9, 10

[KÖP13] Kurt M., Öztürk A., Peers P.: A compact Tucker-based factorization model for heterogeneous subsurface scattering. In *Theory and Practice of Computer Graphics (TPCG)* (2013), pp. 85–92. doi:10.2312/LocalChapterEvents.TPCG.TPCG13.085-092. 3, 7

[Kur21] Kurt M.: GenSSS: a genetic algorithm for measured subsurface scattering representation. *The Visual Computer 37*, 2 (2021), 307–323. doi:10.1007/s00371-020-01800-0. 2, 3

[LGF*22] Li Q., Guo J., Fei Y., Li F., Guo Y.: NeuLighting: Neural lighting for free viewpoint outdoor scene relighting with unconstrained photo collections. In *SIGGRAPH Asia 2022 Conference Papers* (2022), pp. 13:1–13:9. doi:10.1145/3550469.3555384. 2

[LHW21] Leonard L., Hoehlein K., Westermann R.: Learning multiple-scattering solutions for sphere-tracing of volumetric subsurface effects. *Computer Graphics Forum 40*, 2 (2021), 165–178. doi:10.1111/cgf.142623. 2

[LTL*22] Lyu L., Tewari A., Leimkühler T., Habermann M., Theobalt C.: Neural radiance transfer fields for relightable novel-view synthesis with global illumination. In *European Conference on Computer Vision (ECCV)* (2022), Springer, pp. 153–169. doi:10.1007/978-3-031-19790-1_10. 2

[MST*21] Mildenhall B., Srinivasan P. P., Tancik M., Barron J. T., Ramamoorthi R., Ng R.: NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM 65*, 1 (2021), 99–106. doi:10.1145/3503250. 2

[NDVZJ19] Nimier-David M., Vicini D., Zeltner T., Jakob W.: Mitsuba 2: a retargetable forward and inverse renderer. *ACM Transactions on Graphics 38*, 6 (2019), 203:1–203:17. doi:10.1145/3355089.3356498. 5

[NRH03] Ng R., Ramamoorthi R., Hanrahan P.: All-frequency shadows using non-linear wavelet lighting approximation. *ACM Transactions on Graphics 22*, 3 (2003), 376–381. doi:10.1145/882262.882280. 5

[PBD*10] Parker S. G., Bigler J., Dietrich A., Friedrich H., Hoberock J., Luebke D., McAllister D., McGuire M., Morley K., Robison A., Stich M.: OptiX: a general purpose ray tracing engine. *ACM Transactions on Graphics 29*, 4 (2010), 66:1–66:13. doi:10.1145/1778765.1778803. 6

[PGM*19] Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., Killeen T., Lin Z., Gimelshein N., Antiga L., Desmaison A., Kopf A., Yang E., DeVito Z., Raison M., Tejani A., Chilamkurthy S., Steiner B., Fang L., Bai J., Chintala S.: PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)* (2019), pp. 8024–8035. doi:10.48550/arXiv.1912.01703. 6

[PH00] Pharr M., Hanrahan P.: Monte Carlo evaluation of non-linear scattering equations for subsurface reflection. In *SIGGRAPH 2000* (2000), ACM, pp. 75–84. doi:10.1145/344779.344824. 3

[PJH23] Pharr M., Jakob W., Humphreys G.: *Physically Based Rendering: From Theory to Implementation*, fourth ed. MIT Press, 2023. 4, 5, 8, 10

[Pre65] Preisendorfer R. W.: *Radiative Transfer on Discrete Spaces*. Pergamon Press, 1965. doi:10.1016/C2013-0-05368-6. 2, 3

[PvBM*06] Peers P., vom Berge K., Matusik W., Ramamoorthi R., Lawrence J., Rusinkiewicz S., Dutré P.: A compact factored representation of heterogeneous subsurface scattering. *ACM Transactions on Graphics 25*, 3 (2006), 746–753. doi:10.1145/1141911.1141950. 2, 3, 7

[RBRD22] Rainer G., Bousseau A., Ritschel T., Drettakis G.: Neural precomputed radiance transfer. *Computer Graphics Forum 41*, 2 (2022), 365–378. doi:10.1111/cgf.14480. 3

[RGJW20] Rainer G., Ghosh A., Jakob W., Weyrich T.: Unified neural encoding of BTFs. *Computer Graphics Forum 39*, 2 (2020), 167–178. doi:10.1111/cgf.13921. 2, 3

[RSB*21] Rittig T., Sumin D., Babaei V., Didyk P., Voloboy A., Wilkie A., Bickel B., Myszkowski K., Weyrich T., Křivánek J.: Neural acceleration of scattering-aware color 3D printing. *Computer Graphics Forum 40*, 2 (2021), 205–219. doi:10.1111/cgf.142626. 3

[RXL*23] Raghavan N., Xiao Y., Lin K.-E., Sun T., Bi S., Xu Z., Li T.-M., Ramamoorthi R.: Neural free-viewpoint relighting for glossy indirect illumination. *Computer Graphics Forum 42*, 4 (2023), e14885:1–e14885:13. doi:10.1111/cgf.14885. 3, 5

[SLB*21] Sun T., Lin K.-E., Bi S., Xu Z., Ramamoorthi R.: NeLF: neural light-transport field for portrait view synthesis and relighting. In *Eurographics Symposium on Rendering (EGSR)* (2021), Eurographics Association, pp. 155–166. doi:10.2312/sr.20211299. 2

[SLS05] Sloan P.-P., Luna B., Snyder J.: Local, deformable precomputed radiance transfer. *ACM Transactions on Graphics 24*, 3 (2005), 1216–1224. doi:10.1145/1073204.1073335. 2

[TSM*20] Tancik M., Srinivasan P., Mildenhall B., Fridovich-Keil S., Raghavan N., Singhal U., Ramamoorthi R., Barron J., Ng R.: Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems (NeurIPS)* (2020), pp. 7537–7547. doi:10.48550/arXiv.2006.10739. 4

[TTJ*24] TG T., Tran D. M., Jensen H. W., Ramamoorthi R., Frisvad J. R.: Neural SSS: Lightweight object appearance representation. *Computer Graphics Forum 43*, 4 (2024), e15158. doi:10.1111/cgf.15158. 3, 11

[VHH*19] Vorba J., Hanika J., Herholz S., Müller T., Křivánek J., Keller A.: Path guiding in production. In *ACM SIGGRAPH 2019 Courses* (2019), ACM, p. Article 18. doi:10.1145/3305366.3328091. 6

[VKJ19] Vicini D., Koltun V., Jakob W.: A learned shape-adaptive subsurface scattering model. *ACM Transactions on Graphics 38*, 4 (2019), 127:1–127:15. doi:10.1145/3306346.3322974. 2

[WBSS04] 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 (2004), 600–612. doi:10.1109/TIP.2003.819861. 7

[WMLT07] Walter B., Marschner S. R., Li H., Torrance K. E.: Microfacet models for refraction through rough surfaces. In *Eurographics Sypmosium on Rendering (EGSR)* (2007), Eurographics Association, pp. 195–206. doi:10.2312/EGWR/EGSR07/195-206. 4

[WTL05] Wang R., Tran J., Luebke D.: All-frequency interactive relighting of translucent objects with single and multiple scattering. *ACM Transactions on Graphics 24*, 3 (2005), 1202–1207. doi:10.1145/1073204.1073333. 2, 5

[WWH*10] Wang Y., Wang J., Holzschuch N., Subr K., Yong J.-H., Guo B.: Real-time rendering of heterogeneous translucent objects with arbitrary shapes. *Computer Graphics Forum 29*, 2 (2010), 497–506. doi:10.1111/j.1467-8659.2009.01619.x. 2

[WZT*08] Wang J., Zhao S., Tong X., Lin S., Lin Z., Dong Y., Guo B., Shum H.-Y.: Modeling and rendering of heterogeneous translucent materials using the diffusion equation. *ACM Transactions on Graphics 27*, 1 (2008), 9:1–9:18. doi:10.1145/1330511.1330520. 2

[XWH*23] Xu B., Wu L., Hasan M., Luan F., Georgiev I., Xu Z., Ramamoorthi R.: NeuSample: importance sampling for neural materials. In *SIGGRAPH 2023 Conference Proceedings* (2023), ACM, pp. 41:1–41:10. doi:10.1145/3588432.3591524. 11

[YZL*22] Yao Y., Zhang J., Liu J., Qu Y., Fang T., McKinnon D., Tsin Y., Quan L.: NeILF: neural incident light field for material and lighting estimation. In *European Conference on Computer Vision (ECCV)* (2022), Springer, pp. 23–27. doi:10.1007/978-3-031-19821-2_40. 2

[ZCD*23] Zeng C., Chen G., Dong Y., Peers P., Wu H., Tong X.: Relighting neural radiance fields with shadow and highlight hints. In *SIGGRAPH 2023 Conference Proceedings* (2023), ACM, pp. 73:1–73:11. doi:10.1145/3588432.3591482. 2

[ZFT*21] Zhang X., Fanello S., Tsai Y.-T., Sun T., Xue T., Pandey R., Orts-Escolano S., Davidson P., Rhemann C., Debevec P., Ramamoorthi R., Freeman W. T.: Neural light transport for relighting and view synthesis. *ACM Transactions on Graphics 40*, 1 (2021), 9:1–9:17. doi:10.1145/3446328. 3

[ZHM*23] Zheng C., Huo Y., Mo S., Zhong Z., Wu Z., Hua W., Wang R., Bao H.: NeLT: object-oriented neural light transfer. *ACM Transactions on Graphics 42*, 5 (2023), 163:1–163:16. doi:10.1145/3596491. 3

[ZRW*23] Zeltner T., Rousselle F., Weidlich A., Clarberg P., Novák J., Bitterli B., Evans A., Davidovič T., Kallweit S., Lefohn A.: Real-time neural appearance models. arXiv:2305.02678 [cs.GR], 2023. doi:10.48550/arXiv.2305.02678. 2

[ZSB*23] Zhu S., Saito S., Bozic A., Aliaga C., Darrell T., Lassner C.: Neural relighting with subsurface scattering by learning the radiance transfer gradient. arXiv:2306.09322 [cs.CV], 2023. doi:10.48550/arXiv.2306.09322. 2

[ZSD*21] Zhang X., Srinivasan P. P., Deng B., Debevec P., Freeman W. T., Barron J. T.: NeRFactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics 40*, 6 (2021), 237:1–237:18. doi:10.1145/3478513.3480496. 2, 7, 9, 10

[ZSH*22] Zhang Y., Sun J., He X., Fu H., Jia R., Zhou X.: Modeling indirect illumination for inverse rendering. In *Computer Vision and Pattern Recognition (CVPR)* (2022), IEEE, pp. 18622–18631. doi:10.1109/CVPR52688.2022.01809. 2

[ZZW*21] Zheng C., Zheng R., Wang R., Zhao S., Bao H.: A compact representation of measured BRDFs using neural processes. *ACM Transactions on Graphics 41*, 2 (2021), 14:1–14:15. doi:10.1145/3490385. 3