

Neural BSSRDF: Object Appearance Representation Including Heterogeneous Subsurface Scattering

Thomson TG¹, Jeppe Revall Frisvad¹, Ravi Ramamoorthi², and Henrik Wann Jensen³

¹Technical University of Denmark

²University of California, San Diego

³Luxion

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 rendering easily becomes computationally heavy. We propose a compact and efficient neural method for representing and rendering the appearance of heterogeneous translucent objects. The neural representation function resembles a bidirectional scattering-surface reflectance distribution function (BSSRDF). However, conventional BSSRDF models assume a planar half-space medium and only surface variation of the material, which is often not a good representation of the appearance of real-world objects. Our method represents the BSSRDF of a full object taking its geometry and 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 all-frequency 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 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.

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 consider each surface and subsurface scattering event; the BSSRDF is a global formulation considering only incident and emergent light [Pre65].

When using a BSSRDF, we most commonly employ an approximate analytic model [JMLH01]. Models exist that generalize to include dependency on the directions of incidence and observation [FD17] or on the local object geometry [VKJ19]. These models are however for rendering of homogeneous translucent materials.

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 [DWPH20]. An often used solution is to approximate the subsurface scattering using a factored BSSRDF representation [PvBM*06; 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 rep-

resentation 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 in PRT models based on analytic BSSRDF models [WTL05].

Another approach to capturing the full appearance of a translucent object is by using a neural radiance field (NeRF) [MST*21]. This is however a representation of the object appearance in one particular lighting environment. Derived methods [BBJ*21; ZSD*21] enable arbitrary incident illumination by jointly optimizing a model for shape, BRDF, and illumination. By assuming a BRDF model, these methods however sacrifice their ability to represent the appearance of a translucent object. A method for relighting of a captured NeRF has been achieved by assuming a specific type of lighting environment [LGF*22], in this case an outdoor sun and sky lighting model was assumed.

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 BSSRDF depends on the full object geometry, not only one point of incidence. In this way, we 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. We compare the accuracy of our model with that of classic appearance representation in spherical harmonics as well as factored NeRF [ZSD*21] and a neural BTF model [KMX*21].

2. Theory

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

$$L_r(\mathbf{x}_o, \vec{\omega}_o) = \int_{A_i} \int_{2\pi} S(\mathbf{x}_i, \vec{\omega}_i; \mathbf{x}_o, \vec{\omega}_o) L_i(\mathbf{x}_i, \vec{\omega}_i) \cos \theta_i d\omega_i dA_i, \quad (1)$$

where L_i is the incident radiance from the direction $\vec{\omega}_i$ at position $\mathbf{x}_i \in A_i$, so that the angle of incidence is θ_i , and S is the BSSRDF. We let A_i 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

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

and we can write up a BSSRDF for distant illumination:

$$\hat{S}(\vec{\omega}_i; \mathbf{x}_o, \vec{\omega}_o) = \int_{A_i} S(\mathbf{x}_i, \vec{\omega}_i; \mathbf{x}_o, \vec{\omega}_o) V(\mathbf{x}_i, \vec{\omega}_i) (\vec{\omega}_i \cdot \vec{n}_i) dA_i, \quad (3)$$

where \vec{n}_i is the surface normal at \mathbf{x}_i . Assuming A_i 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

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

With an MLP representation of \hat{S} , we can use this equation for computing reflected radiance directly as a shader in a rendering system.

2.1. Neural BSSRDF

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 directional lights of unit radiance. A directional light of direction $\vec{\omega}_e$ illuminates the object in each image. Letting $\mathbf{x}_{o,k}, \vec{\omega}_{o,k}$ denote the position and direction of observation for a ray through pixel k , we find parameters ψ^* for the MLP to represent the observation by

$$\psi^* = \arg \min_{\psi} \mathbb{E}_k [(f_\psi(-\vec{\omega}_e; \mathbf{x}_{o,k}, \vec{\omega}_{o,k}) - L_{r,k})^2], \quad (5)$$

where $L_{r,k} \approx \hat{S}(-\vec{\omega}_e; \mathbf{x}_{o,k}, \vec{\omega}_{o,k})$ 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_{ψ^*} will serve as a representation of \hat{S} in Eq. 4.

Our concept is a variation of neural PRT [RBRD22], but we do not separate our model into a single environment encoded in spherical harmonics 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.

The idea of using a factored transport in PRT is similar to our method. Our representation, like theirs, is decomposed into two components, specular and subsurface. We use traditional algorithms for the specular component and our neural BSSRDF for the subsurface scattering approximation.

2.2. Importance Sampling

When using our model in a renderer, we evaluate Eq. 4 with our neural BSSRDF f_{ψ^*} in place of \hat{S} . For Monte Carlo integration of Eq. 4, we sample directions of incidence $\vec{\omega}_i$ according to a probability density function $\text{pdf}(\vec{\omega}_i)$. The N th sample estimator of the integral is then

$$L_r(\mathbf{x}_o, \vec{\omega}_o) \approx \frac{1}{N} \sum_{j=1}^N \frac{L_i(\vec{\omega}_{i,j}) f_{\psi^*}(\vec{\omega}_{i,j}; \mathbf{x}_o, \vec{\omega}_o)}{\text{pdf}(\vec{\omega}_{i,j})}, \quad (6)$$

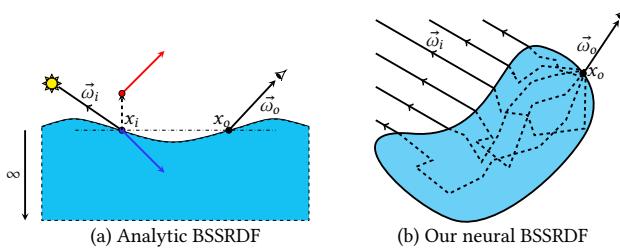


Figure 1: The technical difference between conventional analytic BSSRDF models and our neural BSSRDF model. Our model estimates the full appearance of an object illuminated by a directional light, in contrast to the conventional assumption of a ray incident on a half-space medium. This enables us to represent objects with intricate heterogeneous surface and subsurface scattering with high accuracy. Conventional models would have to violate many assumptions to represent such objects.

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], for example. 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 \vec{x}_o . As an alternative, we would like an importance sampling technique with a pdf roughly proportional to the value of the neural BSSRDF \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 spherical coordinates (θ_i, ϕ_i) and a normalized 2D Gaussian kernel with standard deviation σ for the pdf. To simplify the importance sampling module, we assume that the normal direction in object space is the most important and use a Gaussian kernel with zero mean. The main difference from importance sampling for a spatially varying BRDF is that light incident from any direction on the unit sphere can be important ($\theta \in [0, \pi]$). We use the Box-Muller transform [HT07] to sample direction vectors according to the σ retrieved from the network.

3. Implementation

Our neural BSSRDF is an MLP representation of \hat{S} (Eq. 3) that takes as input $\vec{\omega}_i, \vec{\omega}_o, \vec{x}_o$ (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 object surface position (see Figure 1). The result is then scaled to the contribution from a distant light to get the contribution in a Monte Carlo renderer. Our neural BSSRDF only captures the subsurface scattering. Surface reflection effects (Fresnel effects) are computed separately.

For a given point of observation \vec{x}_o , light incident along the normal \vec{n}_o would often have high importance when we only consider subsurface scattering. This is due to Fresnel reflectance increasing with increasing angle of incidence so that less light refracts into the medium for more oblique incidence. We thus define our

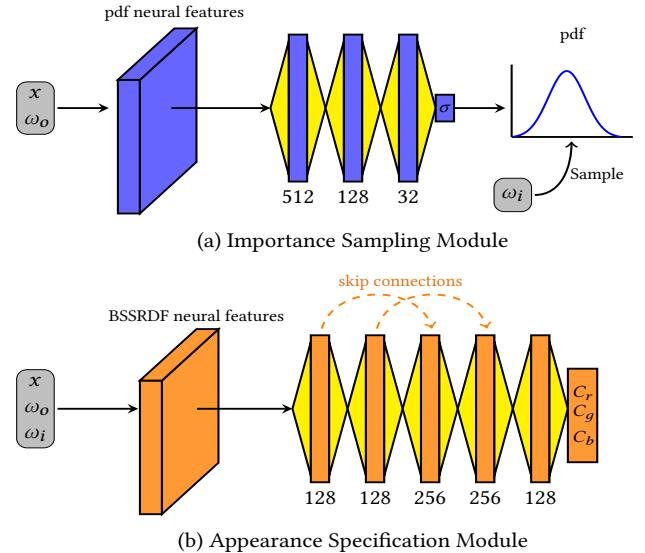


Figure 2: Our Network Architecture consists of 2 parts: Importance Sampling Module and Appearance Specification Module.

Gaussian lobe in the tangent space of \vec{x}_o and make it symmetric around the normal direction. To improve the expressivity of the function, we use the symmetric generalized Gaussian distribution function [VA89]. In practice, we uniformly sample 32×64 directions (32 inclination θ_i by 64 azimuthal ϕ_i angles) across the unit sphere for every position and direction of observation $(\vec{x}_o, \vec{\omega}_o)$ in the neural BSSRDF, see Figure 2. We convert this into a distribution and minimise the distance between the generated distribution and our pdf representation.

3.1. Network Architecture

Our architecture consists of two major components: an importance sampling module and an appearance specification module, see Figure 2. Due to the known inability of MLPs to learn high frequency functions in low dimensional domains [TSM^{*}20] and due to the bias towards low-frequency functions, we cast our problem into a higher dimensional space before passing it to the MLP. Both the modules therefore have two components: (a) feature extraction and (b) neural network. The feature extraction module is similar in both the modules where we use Fourier feature maps. Our Fourier feature map is defined by

$$f(x) = [\cos(2\pi Gx), \sin(2\pi Gx)]^T, \quad (7)$$

where G is a matrix of size $256 \times n$ representing a 2D Gaussian distribution with a mean of $\mu = 18$ and standard deviation of $\sigma = 1$. Note that being a randomly sampled isotropic distribution, the Gaussian G performs frequency modulation on the feature map. Furthermore, the standard deviation (σ) is a significant factor in extracting the features, however $\sigma = 1$ 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 with ReLU

(Rectified Linear Unit) activations between each layer. This module is responsible for predicting the standard deviation (σ) for the Gaussian distribution on the appearance specification module. Since the aim is to reduce the difference of two distributions, we use Kullback-Leibler divergence (KLD) loss for fitting our distribution prediction. Furthermore, we normalize the spherical coordinates for out of bound values while drawing samples.

The appearance specification module, on the other hand, has a 5-layers MLP with 128 neurons each in every layer with two skip connections: from layer 1 to 3 and layer 2 to 4, respectively. The skip connections reduce the memory footprint of the network to approximately 649KB on disk, where the weights are well optimized at every layer in the neural architecture. We used ReLU as activation after every layer, except for the last layer after which we used no activation, and we used L2 as the loss function between our prediction and ground truth.

3.2. 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 rotated and translated at a uniform distance from the center of the geometry. To guarantee convergence, each rendered image had a high sample count. We also filtered the data to only include ray hits on the geometry.

We use an online learning method to train our model for the appearance specification module. In this method, the renderer and the neural network run in parallel. The task of the renderer is to keep rendering data at all times. When new data becomes available, the renderer pushes it into the 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. Another approach to do the same would be an offline method which essentially means that we collect and store the necessary data into the disk and train the model independently. This can be done with a general purpose renderer like Mitsuba [NVZJ19] or PBRT [PJH23]. For quantitative analysis, we have tested an offline learning method where we collect data from a renderer and train the model as a secondary step. We keep 5% of the data as test set and another 5% was kept for validation. We found that the collected data $\hat{S}(\vec{\omega}_i; \mathbf{x}_o, \vec{\omega}_o)$ from approximately ~ 835 images was adequate to have the network represent a heterogeneous medium. Note that the images do not directly correspond to the amount of data acquired. Every view might have a different number of hits on the geometry.

The importance sampling module is trained on top of the appearance specification module. Thus, it is considered to be in the post-processing stage, and since it is trained on inference data from the appearance specification module, no explicit data are generated to train this module. We randomly take the position \mathbf{x}_o on the geometry, $\vec{\omega}_o$ on the hemisphere along with a batch of $32 * 64$ uniformly distributed $\vec{\omega}_i$ on the sphere. The resultant data are collected into bins which form our ground truth distribution. This distribution is expected to be approximately Gaussian (discussed in Sec. 2.2), why we fit a network that predicts the standard deviation (σ) of the resultant distribution.

3.3. Precomputed Radiance Transfer (PRT)

By employing a straightforward MLP, we demonstrate that our neural representation can be executed interactively on a modern graphics processing unit (GPU). Furthermore, our approach involves simplifying the traditional BSSRDF by reducing its dimensionality to be on par with a BRDF function. This makes our model extensible to a wide range of applications. To demonstrate the effectiveness of our methodology, we use a spherical harmonics-based PRT technique and obtain real-time rendering. We consider this a variant of our material representation, which achieves real-time performance at the cost of accuracy. 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 the lower dimensionality requirements, PRT has not so far been very effective in addressing higher dimensional problems, such as a BSSRDF.

The lighting coefficients are computed at run-time in the standard way,

$$b_l^m = \int_{4\pi} L_i(\vec{\omega}_i) Y_l^m(\vec{\omega}_i) d\omega_i, \quad (8)$$

where L_i is the incident environment map and Y_l^m are the (real) spherical harmonic basis functions.

We also precompute the spherical harmonic coefficients of the light transport in terms of the neural BSSRDF,

$$a_l^m(\mathbf{x}_o, \vec{\omega}_o) = \int_{4\pi} \hat{S}(\vec{\omega}_i; \mathbf{x}_o, \vec{\omega}_o) Y_l^m(\vec{\omega}_i) d\omega_i, \quad (9)$$

where the subsurface scattering function \hat{S} is integrated with the (real) spherical harmonic Y_l^m and we use the neural BSSRDF MLP model f_{Ψ^*} for \hat{S} . We use degree 2 spherical harmonics $l \leq 2$ with 9 coefficients to represent our coefficients a_l^m and b_l^m . In addition to its lower dimensionality, our neural BSSRDF enables inexpensive low variance estimates of the integral above for coefficients a_l^m . In contrast, path tracing does not provide reliable low variance estimates, often necessitating an exceptionally high number of samples to achieve an acceptable level of variance.

It is important to highlight that the computation of a_l^m occurs at each vertex within the object geometry for every linearly distributed angle $\vec{\omega}_o$ 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 as:

$$L_r(\mathbf{x}_o, \vec{\omega}_o) = \sum_{l=0}^{\infty} \sum_{m=-l}^l a_l^m(\mathbf{x}_o, \vec{\omega}_o) b_l^m. \quad (10)$$

It is important to note that a spherical harmonic representation has inherent limitations in accurately representing high-frequency functions. As a consequence, the performance gains achieved through this conversion are accompanied by a trade-off in accuracy.

In order to surpass the limitations of spherical harmonics in capturing intricate details with precision, it is possible to explore alternative representations like wavelets. Wavelets are well-known for their superior ability to represent high-frequency functions, but

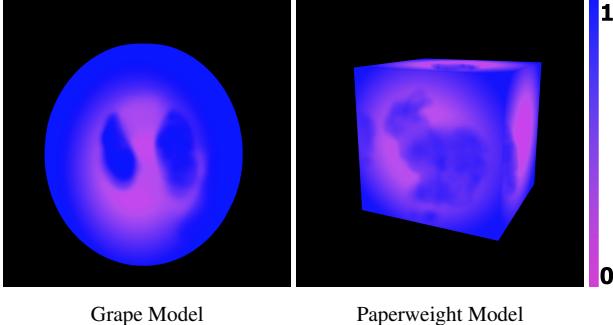


Figure 3: Visualizations of the standard deviation σ predicted by our importance sampling module. The standard deviation is generated once per pixel and subsequently $\vec{\omega}_i$ is sampled for the appearance specification module using the generated distribution.

capturing full view and light dependent effects remains a significant challenge. Utilizing wavelets, or similar representations can achieve improved fidelity with better accuracy in representing complex materials that exhibit intricate high-frequency behavior.

3.4. Framework and Rendering

During network training, we employ our in-house 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, for inference, weights from the PyTorch model are extracted and computed on our implementation of network inference. Our framework can be easily integrated into Monte Carlo renderers since we do not batch queries at inference time while each shading point is evaluated independently.

Rendering is done inside a shader, where our importance sampling module is used for inference. This returns the standard deviation (σ) of the expected Gaussian pdf, see examples in Figure 3. A sample direction $\vec{\omega}_{i,j}$ is drawn from the predicted Gaussian pdf. Our appearance specification module is inferred for a relative contribution based on the inputs (i.e. $\vec{\omega}_o, \vec{x}, \vec{\omega}_i$), which in turn is scaled for the actual contribution from the distant illumination. Furthermore, surface reflection (Fresnel effect) is computed separately.

In our PRT simulation, as part of our alternative material representation, we evenly distribute 16×64 $\vec{\omega}_o$ directions at each vertex in object space. Subsequently, we perform a precomputation step to calculate spherical harmonic coefficients for each $\vec{\omega}_o$ by employing Monte Carlo sampling of $\vec{\omega}_i$ at every vertex on the shading geometry. These coefficients are then used for inference of our appearance specification module and construct a signal representing the subsurface scattering phenomenon. Similarly, during the precomputation phase, we compute spherical harmonic coefficients for environment lighting. When it comes to rendering, the coefficients are convolved together to get the desired output.

4. Results

In this section, we demonstrate the ability of our network to faithfully represent and render heterogeneous translucent objects under

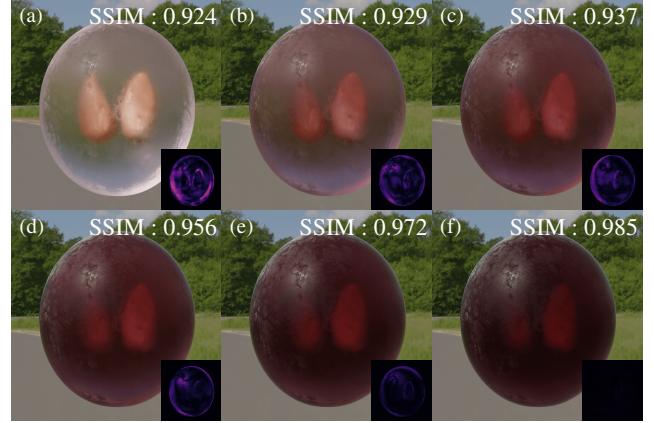


Figure 4: Our method applied to objects with different levels of translucency. The insert at the bottom right shows the \mathcal{FLIP} mean between the images rendered using path tracing and our neural BSSRDF. The SSIM (top-right) demonstrates high accuracy compared to the ground truth.

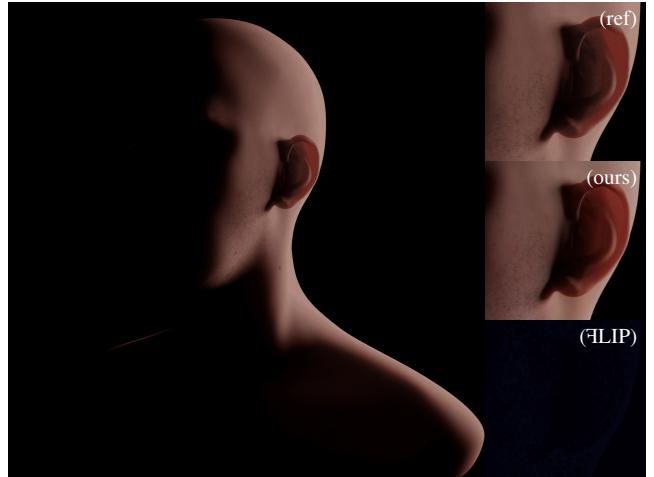


Figure 5: The ability of our method to represent the self-shadowing of a translucent object with non-trivial surface geometry. The back-lighting of the head makes the ear cast a shadow while light is also bleeding through the ear. This example also demonstrates that the neural BSSRDF works for a single directional light just as well as for the environment light used in the other examples. The \mathcal{FLIP} error of the close-up on the ear shows high accuracy as compared with the path traced reference image.

distant illumination. Figure 4 is a test of our model using objects with different levels of translucency illuminated by an environment. In Figure 5, we test an example of just a single directional light illuminating an object with non-trivial surface geometry leading to self-shadowing of the translucent object. A performance and accuracy analysis is provided for five different objects in Table 1. We employ three objective full image quality metrics: mean-squared error (MSE), structural similarity (SSIM) index [WBSS04], and the mean value of \mathcal{FLIP} [ANA*20]. Our network achieves high fidelity

Table 1: Performance and accuracy analysis for our method.

Experiment Name	Model	SPP	Render Time (sec.) \downarrow	Storage (MB) \downarrow	MSE \downarrow	SSIM \uparrow	FLIP mean \downarrow
Grape	Path Tracing	10000	1846.44				
	Ours(Net)	256	2.038	1.31	$1.6 \cdot 10^{-4}$	0.975	0.051
	Ours(SH)	1	$3.2 \cdot 10^{-4}$	21.1	$2.8 \cdot 10^{-3}$	0.983	0.082
Dragon	Path Tracing	10000	2412.56				
	Ours(Net)	256	1.813	1.31	$1.4 \cdot 10^{-5}$	0.990	0.009
	Ours(SH)	1	$1.2 \cdot 10^{-3}$	779.6	$6.2 \cdot 10^{-4}$	0.971	0.045
Paperweight	Path Tracing	10000	2503.83				
	Ours(Net)	256	3.950	3.77	$1.1 \cdot 10^{-3}$	0.987	0.079
	Ours(SH)	1	$3.6 \cdot 10^{-4}$	46.1	$3.4 \cdot 10^{-3}$	0.953	0.115
Drink with ice	Path Tracing	10000	1496.24				
	Ours(Net)	256	3.231	1.31	$3.0 \cdot 10^{-4}$	0.870	0.043
	Ours(SH)	1	$2.8 \cdot 10^{-4}$	19.2	$1.4 \cdot 10^{-3}$	0.969	0.070
Lucy	Path Tracing	10000	1198.36				
	Ours(Net)	256	2.189	0.65	$4.7 \cdot 10^{-4}$	0.986	0.035
	Ours(SH)	1	$3.7 \cdot 10^{-4}$	64.2	$1.1 \cdot 10^{-3}$	0.984	0.382

(FLIP below 0.1), enabling us to represent the complex appearance of the objects while efficiently storing the data. Additionally, our PRT-based implementation for our translucent object appearance representation achieves real-time rendering frame rates (less than 0.01 seconds per frame). 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.

The setup and models in the different scenes that we used in our experiments are as follows:

- *Grape* is a model consisting of a scattering volume behind a refractive interface with surface roughness defined by a texture map. The volume contains two seeds that are diffuse with reflectance defined by an image texture.
- *Dragon* is the XYZ RGB dragon from the 3D scanning repository of the Stanford Computer Graphics Laboratory (<http://graphics.stanford.edu/data/3Dscanrep/>). 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].

Ablation study. We conducted an ablation test by evaluating the performance of various network sizes in comparison to a 7-layered network, see Figure 6. Our findings indicate that a 7-layered network is generally sufficient for representing most types of heterogeneous translucent objects. It is worth noting that we incorporate skip connections between alternate layers as our tests have shown

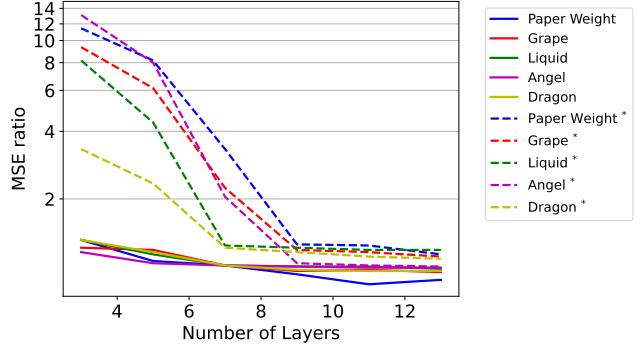


Figure 6: The influence of varying the number of layers in our *appearance specification module* (Figure 2b) on the resulting mean squared error (MSE) taken relative to a 7-layers MLP with skip connections. The asterisk * marks the same results but without skip connections. This clearly demonstrates the importance of the skip connections and illustrates the trade-off in selecting the number of layers (we chose 5).

that this architectural design choice leads to improved performance with better compression.

Low storage. During the design of our architecture for material representation, we considered storage efficiency important. Comparing with other techniques for storing heterogeneous scattering volumes, such as the approach proposed by Kurt et al. [KÖP13], our model achieves a more compact representation. However, since we did not have access to the same dataset as Kurt et al., a direct quantitative comparison was not possible. Table 1 provides evidence of the good 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 to were spent to represent measured heterogeneous translucent materials).

Quality of representations. To provide the reader with an opportunity to qualitatively assess the ability of our method to represent object appearance, Figure 7 provides a comparison of our different types of BSSRDF representations with path traced reference images. We specifically compare use of our appearance specification module with environment importance sampling [PJH23] as well as use of our own importance sampling module. The scenes are the same as in the quantitative analysis in Table 1.

When using environment importance sampling, we may sample directions that are not very relevant to our appearance representation. In contrast, our importance sampling module puts more focus on our appearance specification module resulting in reduced variance at equal sample counts. To ensure a fair comparison, we render the images at 256 samples per pixel with and without our importance sampling module. If an object is placed in a high dynamic range (HDR) environment with large variation of intensities, or in a scene with an idealized directional source, our importance sampling would not necessarily be better. In general, multiple importance sampling would be desirable, and a network can be used to find multiple importance sampling weights [FWH*22].

Additionally, we compare with our implementation of the spherical harmonic (SH) representation of our neural BSSRDF to demonstrate a notable performance improvement at the expense of accuracy. While the FLIP error rates is higher in the SH representation compared to our appearance specification module, the objects seem to maintain an acceptable visual fidelity. For a visual comparison, we refer the reader to the supplementary video which demonstrates interactive rendering of our network representation and real-time rendering of our spherical harmonic representation.

Comparison with other methods. Table 2 provides a quantitative analysis and Figure 8 enables a qualitative assessment of the accuracy and performance of our neural BSSRDF as compared with other neural object and material representation methods. In the absence of a dedicated neural BSSRDF representation technique, we conducted a comparison with NeRFactor [ZSD*21] and NeuMip [KMX*21]. The selection of NeRFactor was motivated by its capacity to infer BRDF properties using a handful of images and effectively relight scenes in unfamiliar lighting conditions. Furthermore, NeuMip stands as a valuable technique for representing BRDF materials. 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.

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 Figure 8. 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

Table 2: Image quality metrics for NeRFactor [ZSD*21], NeuMip [KMX*21], and the network version of our method when comparing with a path traced reference image. The values were computed for rendered images of resolution 1920^2 (MSE, SSIM1, FLIP). The SSIM metric prefers blur to noise. If we downsample the images to 480^2 , the SSIM values change significantly (SSIM2).

Experiment	Metric	NeRFactor	NeuMip	Ours(Net)
Grape (outdoors)	MSE \downarrow	$3.34 \cdot 10^{-3}$	$1.82 \cdot 10^{-4}$	$9.28 \cdot 10^{-5}$
	SSIM1 \uparrow	$8.86 \cdot 10^{-1}$	$9.82 \cdot 10^{-1}$	$9.49 \cdot 10^{-1}$
	SSIM2 \uparrow	$8.89 \cdot 10^{-1}$	$9.90 \cdot 10^{-1}$	$9.98 \cdot 10^{-1}$
	$\text{FLIP}\downarrow$	$1.32 \cdot 10^{-1}$	$5.12 \cdot 10^{-2}$	$1.74 \cdot 10^{-2}$
Grape (indoors)	MSE \downarrow	$2.84 \cdot 10^{-3}$	$9.81 \cdot 10^{-4}$	$3.11 \cdot 10^{-4}$
	SSIM1 \uparrow	$9.12 \cdot 10^{-1}$	$9.79 \cdot 10^{-1}$	$8.70 \cdot 10^{-1}$
	SSIM2 \uparrow	$9.08 \cdot 10^{-1}$	$9.83 \cdot 10^{-1}$	$9.94 \cdot 10^{-1}$
	$\text{FLIP}\downarrow$	$1.37 \cdot 10^{-1}$	$1.60 \cdot 10^{-1}$	$3.01 \cdot 10^{-2}$
Paperweight (outdoors)	MSE \downarrow	$9.31 \cdot 10^{-3}$	$7.40 \cdot 10^{-4}$	$6.91 \cdot 10^{-4}$
	SSIM1 \uparrow	$9.19 \cdot 10^{-1}$	$9.73 \cdot 10^{-1}$	$7.57 \cdot 10^{-1}$
	SSIM2 \uparrow	$8.03 \cdot 10^{-1}$	$9.43 \cdot 10^{-1}$	$9.74 \cdot 10^{-1}$
	$\text{FLIP}\downarrow$	$1.63 \cdot 10^{-1}$	$6.08 \cdot 10^{-2}$	$2.79 \cdot 10^{-2}$
Paperweight (indoors)	MSE \downarrow	$1.01 \cdot 10^{-2}$	$1.95 \cdot 10^{-3}$	$1.15 \cdot 10^{-3}$
	SSIM1 \uparrow	$9.14 \cdot 10^{-1}$	$9.60 \cdot 10^{-1}$	$7.54 \cdot 10^{-1}$
	SSIM2 \uparrow	$8.07 \cdot 10^{-1}$	$9.14 \cdot 10^{-1}$	$9.71 \cdot 10^{-1}$
	$\text{FLIP}\downarrow$	$1.70 \cdot 10^{-1}$	$8.90 \cdot 10^{-2}$	$4.91 \cdot 10^{-2}$

learned. Additionally, the trilinear interpolation used for the texture pyramid introduces blur, particularly in high-frequency details. Our neural BSSRDF approach addresses this issue by incorporating feature extraction through a Fourier transform, allowing the network to learn and capture high-frequency details more effectively. This can be seen in the third column of our tests in Figure 8.

In Table 2, the SSIM value for our network representation was lower than for NeuMip. This is due to SSIM preferring 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 Figure 7), 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.

5. Discussion

Importance sampling Our importance sampling technique uses 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 limitations such as non-representativeness of the distribution function. One outlier in a Gaussian distribution could be that sharp edges on the object geometry may require a specific angle closer to grazing to be more important than nearby angles. As the Gaussian distribution is a mono-

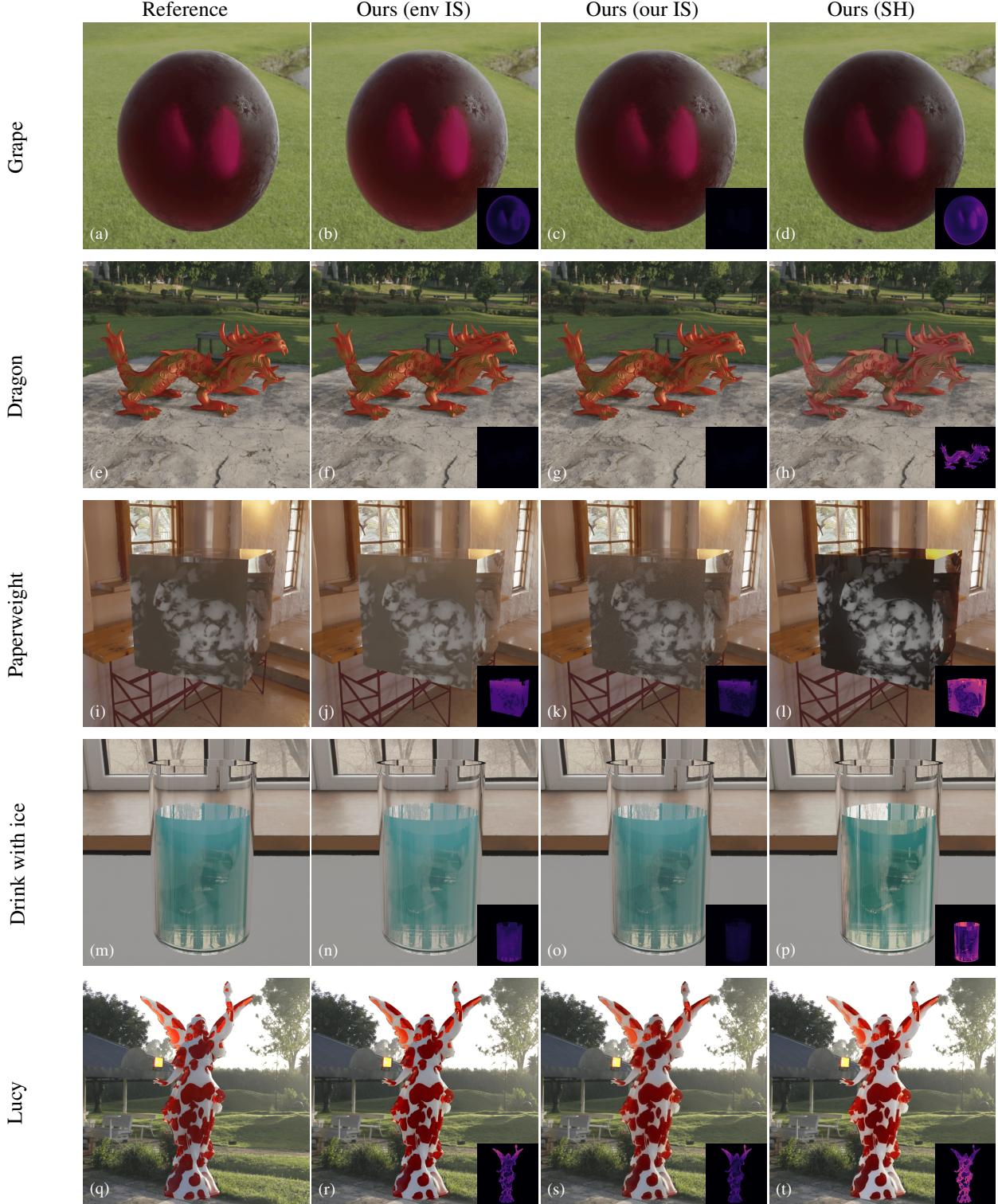


Figure 7: The ability of different versions of our model to represent path traced reference images. We show traditional importance sampling of the environment map [PJH23] (env IS) compared against our importance sampling technique described in Sec. 2.2 (our IS) and the spherical harmonic representation of our model for real-time rendering described in Sec. 3.3 (SH). Each image is rendered at 256 samples per pixel (SPP). The bottom right corners of the images show the flip error metric.

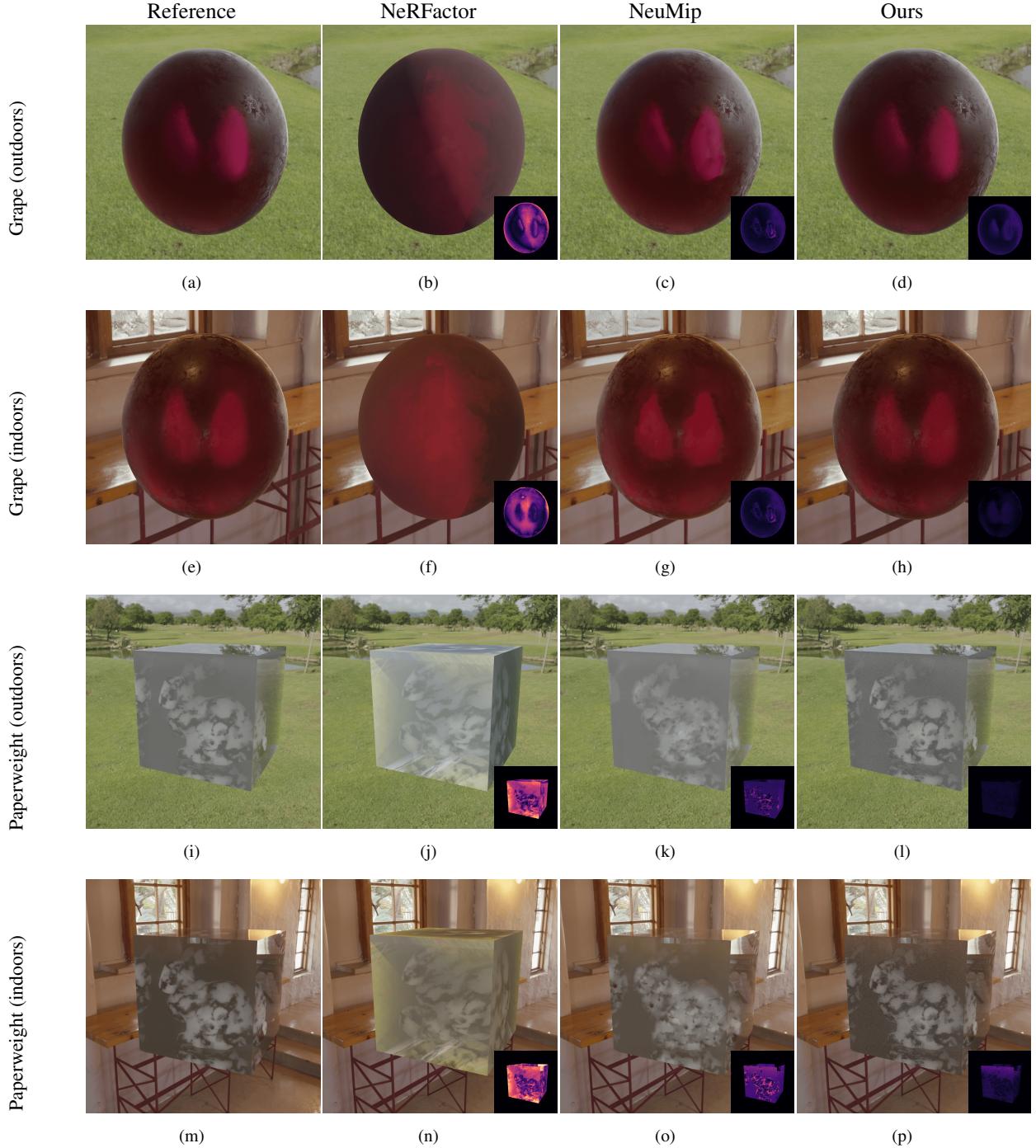


Figure 8: Comparison with other neural methods for object or material appearance capture. We compare with NeRFactor [ZSD*21], which is a BRDF approximation that enables change of the lighting environment, and NeuMip [KMX*21], a texture-based material representation method. The bottom right corner of the images show the flip error metric.

tonically decreasing function, it may not adequately represent such sharp changes. A sum of Gaussians might be preferable.

Limitations Our model assumes distant illumination and is thus limited regarding representation of local illumination. If we were to learn the full BSSRDF function of a heterogeneous translucent object, rendering would become more expensive as we would need to Monte Carlo integrate contributions from light incident across the object surface at render time. We believe it is possible, but the method would be significantly more computationally expensive at render time due to the many required network evaluations. 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 so far.

6. Conclusion

We have presented a method for representing the appearance of heterogeneous translucent objects illuminated by an arbitrary distant lighting environment. Our approach leverages a MLP to approximate a relative contribution metric across a 7-dimensional function with two dimensions for each of the incoming and outgoing light directions, and another three dimensions for the spatial position. By incorporating the capability of importance sampling the object, we have developed a systematic approach to represent a full-object BSSRDF for the case of arbitrary distant illumination.

This work represents a significant advancement in utilizing learned function approximation techniques to alleviate the computational complexities associated with physically based rendering that involves subsurface scattering. It serves as a step towards enabling more efficient and accurate rendering of translucent objects through the adoption of lightweight neural networks.

ACKNOWLEDGEMENTS

This research is a part of PRIME which is funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska Curie grant agreement No 956585.

This work was supported in part by NSF grant 2212085, gifts from Adobe, the Ronald L. Graham Chair and the UC San Diego Center for Visual Computing.

References

- [ANA*20] ANDERSSON, PONTUS, NILSSON, JIM, AKENINE-MÖLLER, TOMAS, OSKARSSON, MAGNUS, ÅSTRÖM, KALLE, and FAIRCHILD, MARK D. “ILIP: A difference evaluator for alternating images”. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 3.2 (2020), 15:1–15:23. DOI: [10.1145/3406183.5](https://doi.org/10.1145/3406183.5).
- [AWB11] ARBREE, ADAM, WALTER, BRUCE, and BALA, KAVITA. “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](https://doi.org/10.1109/TVCG.2010.117).
- [BBJ*21] BOSS, MARK, BRAUN, RAPHAEL, JAMPAANI, VARUN, BARRON, JONATHAN T., LIU, CE, and LENSCHE, HENDRIK. “NeRD: Neural reflectance decomposition from image collections”. *ICCV*. 2021, 12684–12694. DOI: [10.1109/ICCV48922.2021.01245](https://doi.org/10.1109/ICCV48922.2021.01245).
- [BDBS22] BELCOUR, LAURENT, DELIOT, THOMAS, BARBIER, WILHELM, and SOLER, CYRIL. “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](https://doi.org/10.1145/3543864).
- [DWWH20] DENG, HONG, WANG, BEIBEI, WANG, RUI, and HOLZSCHUCH, NICOLAS. “A practical path guiding method for participating media”. *Computational Visual Media* 6 (2020), 37–51. DOI: [10.1007/s41095-020-0160-1](https://doi.org/10.1007/s41095-020-0160-1).
- [FD17] FREDERICKX, ROALD and DUTRÉ, PHILIP. “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](https://doi.org/10.1145/3072959.3073681).
- [FJM*20] FRISVAD, JEPPE REVALL, JENSEN, SØREN A., MADSEN, JONAS SKOVLUND, CORREIA, ANTÓNIO, YANG, LI, GREGERSEN, SØREN KIMMER SCHOU, MEURET, YOURI, and 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](https://doi.org/10.1111/cgf.14023).
- [FWH*22] FAN, JIAHUI, WANG, BEIBEI, HASAN, MILOS, YANG, JIAN, and YAN, LING-QI. “Neural layered BRDFs”. *SIGGRAPH 2022 Conference Proceedings*. ACM. 2022, 4:1–4:8. DOI: [10.1145/3528233.3530732](https://doi.org/10.1145/3528233.3530732).
- [HT07] HOWES, LEE and THOMAS, DAVID. “Efficient random number generation and application using CUDA”. *GPU Gems 3*. NVIDIA/Pearson, 2007, 805–830.
- [JMLH01] JENSEN, HENRIK WANN, MARSCHNER, STEPHEN R., LEVOY, MARC, and HANRAHAN, PAT. “A practical model for subsurface light transport”. *SIGGRAPH 2001*. ACM, 2001, 511–518. DOI: [10.1145/383259.383319](https://doi.org/10.1145/383259.383319).
- [KMX*21] KUZNETSOV, ALEXANDR, MULLIA, KRISHNA, XU, ZEXI-ANG, HAŠAN, MILOŠ, and RAMAMOORTHI, RAVI. “NeuMIP: Multi-resolution neural materials”. *ACM Transactions on Graphics* 40.4 (2021), 175:1–175:13. DOI: [10.1145/3450626.3459795](https://doi.org/10.1145/3450626.3459795).
- [KÖP13] KURT, MURAT, ÖZTÜRK, AYDIN, and PEERS, PIETER. “A compact Tucker-based factorization model for heterogeneous subsurface scattering”. *Theory and Practice of Computer Graphics (TPCG)*. 2013, 85–92. DOI: [10.2312/LocalChapterEvents.TPCG.TPCG13.085-092](https://doi.org/10.2312/LocalChapterEvents.TPCG.TPCG13.085-092).
- [Kur21] KURT, MURAT. “GenSSS: a genetic algorithm for measured subsurface scattering representation”. *The Visual Computer* 37.2 (2021), 307–323. DOI: [10.1007/s00371-020-01800-0](https://doi.org/10.1007/s00371-020-01800-0).
- [LGF*22] LI, QUEWEI, GUO, JIE, FEI, YANG, LI, FEICHAO, and GUO, YANWEN. “NeuLighting: Neural lighting for free viewpoint outdoor scene relighting with unconstrained photo collections”. *SIGGRAPH Asia 2022 Conference Papers*. 2022, 13:1–13:9. DOI: [10.1145/3550469.3555384](https://doi.org/10.1145/3550469.3555384).
- [MST*21] MILDENHALL, BEN, SRINIVASAN, PRATUL P., TANCIK, MATTHEW, BARRON, JONATHAN T., RAMAMOORTHI, RAVI, and NG, REN. “NeRF: Representing scenes as neural radiance fields for view synthesis”. *Communications of the ACM* 65.1 (2021), 99–106. DOI: [10.1145/3503250](https://doi.org/10.1145/3503250).
- [NVZJ19] NIMIER-DAVID, MERLIN, VICINI, DELIO, ZELTNER, TIZIAN, and JAKOB, WENZEL. “Mitsuba 2: A retargetable forward and inverse renderer”. *ACM Transactions on Graphics* 38.6 (2019), 203:1–203:17. DOI: [10.1145/3355089.3356498](https://doi.org/10.1145/3355089.3356498).
- [PBD*10] PARKER, STEVEN G., BIGLER, JAMES, DIETRICH, ANDREAS, FRIEDRICH, HEIKO, HOBEROCK, JARED, LUEBKE, DAVID, MCALLISTER, DAVID, MCGUIRE, MORGAN, MORLEY, KEITH, ROBISON, AUSTIN, and STICH, MARTIN. “OptiX: A general purpose ray tracing engine”. *ACM Transactions on Graphics* 29.4 (July 2010), 66:1–66:13. DOI: [10.1145/1778765.1778803](https://doi.org/10.1145/1778765.1778803).

- [PGM*19] PASZKE, ADAM, GROSS, SAM, MASSA, FRANCISCO, et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. *Advances in Neural Information Processing Systems (NeurIPS)*. 2019, 8024–8035. DOI: [10.48550/arXiv.1912.01703](https://doi.org/10.48550/arXiv.1912.01703) 5.
- [PH00] PHARR, MATT and HANRAHAN, PAT. “Monte Carlo evaluation of non-linear scattering equations for subsurface reflection”. *SIGGRAPH 2000*. ACM, 2000, 75–84. DOI: [10.1145/344779.344824](https://doi.org/10.1145/344779.344824) 2.
- [PJH23] PHARR, MATT, JAKOB, WENZEL, and HUMPHREYS, GREG. *Physically Based Rendering: From Theory to Implementation*. fourth edition. MIT Press, 2023 3, 4, 7, 8.
- [Pre65] PREISENDORFER, RUDOLPH W. *Radiative Transfer on Discrete Spaces*. Pergamon Press, 1965. DOI: [10.1016/C2013-0-05368-6](https://doi.org/10.1016/C2013-0-05368-6) 1, 2.
- [PvBM*06] PEERS, PIETER, vom BERGE, KARL, MATUSIK, WOJCIECH, RAMAMOORTHI, RAVI, LAWRENCE, JASON, RUSINKIEWICZ, SZYMON, and DUTRÉ, PHILIP. “A compact factored representation of heterogeneous subsurface scattering”. *ACM Transactions on Graphics* 25.3 (July 2006), 746–753. DOI: [10.1145/1141911.1141950](https://doi.org/10.1145/1141911.1141950) 1, 7.
- [RBRD22] RAINER, GILLES, BOUSSEAU, ADRIEN, RITSCHEL, TOBIAS, and DRETTAKIS, GEORGE. “Neural precomputed radiance transfer”. *Computer Graphics Forum* 41.2 (2022), 365–378. DOI: [10.1111/cgf.14480](https://doi.org/10.1111/cgf.14480) 2.
- [RGJW20] RAINER, GILLES, GHOSH, ABHIJEET, JAKOB, WENZEL, and WEYRICH, TIM. “Unified neural encoding of BTFs”. *Computer Graphics Forum* 39.2 (2020), 167–178. DOI: [10.1111/cgf.13921](https://doi.org/10.1111/cgf.13921) 1.
- [SLS05] SLOAN, PETER-PIKE, LUNA, BEN, and SNYDER, JOHN. “Local, deformable precomputed radiance transfer”. *ACM Transactions on Graphics* 24.3 (2005), 1216–1224. DOI: [10.1145/1073204](https://doi.org/10.1145/1073204) . 1073335 1.
- [TSM*20] TANCIK, MATTHEW, SRINIVASAN, PRATUL, MILDENHALL, BEN, FRIDOVICH-KEIL, SARA, RAGHAVAN, NITHIN, SINGHAL, UTKARSH, RAMAMOORTHI, RAVI, BARRON, JONATHAN, and NG, REN. “Fourier features let networks learn high frequency functions in low dimensional domains”. *Advances in Neural Information Processing Systems (NeurIPS)* (2020), 7537–7547. DOI: [10.48550/arXiv.2006.10739](https://doi.org/10.48550/arXiv.2006.10739) 3.
- [VA89] VARANASI, MAHESH K and AAZHANG, BEHNAAM. “Parametric generalized Gaussian density estimation”. *The Journal of the Acoustical Society of America* 86.4 (1989), 1404–1415. DOI: [10.1121/1.398700](https://doi.org/10.1121/1.398700) 3.
- [VKJ19] VICINI, DELIO, KOLTUN, VLADLEN, and JAKOB, WENZEL. “A learned shape-adaptive subsurface scattering model”. *ACM Transactions on Graphics* 38.4 (2019), 127:1–127:15. DOI: [10.1145/3306346.3322974](https://doi.org/10.1145/3306346.3322974) 1.
- [WBSS04] WANG, ZHOU, BOVIK, ALAN C., SHEIKH, HAMID R., and SIMONCELLI, EERO 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](https://doi.org/10.1109/TIP.2003.819861) 5.
- [WTL05] WANG, RUI, TRAN, JOHN, and LUEBKE, DAVID. “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](https://doi.org/10.1145/1073204.1073333) 2.
- [WWH*10] WANG, YAJUN, WANG, JIAPING, HOLZSCHUCH, NICOLAS, SUBR, KARTIC, YONG, JUN-HAI, and GUO, BAINING. “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](https://doi.org/10.1111/j.1467-8659.2009.01619.x) 1.
- [WZT*08] WANG, JIAPING, ZHAO, SHUANG, TONG, XIN, LIN, STEPHEN, LIN, ZHOUCHE, DONG, YUE, GUO, BAINING, and SHUM, HEUNG-YEUNG. “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](https://doi.org/10.1145/1330511.1330520) 1.
- [ZRW*23] ZELTNER, TIZIAN, ROUSSELLE, FABRICE, WEIDLICH, ANDREA, CLARBERG, PETRIK, NOVÁK, JAN, BITTERLI, BENEDIKT, EVANS, ALEX, DAVIDOVIĆ, TOMÁŠ, KALLWEIT, SIMON, and LEFOHN, AARON. *Real-Time Neural Appearance Models*. Tech. rep. NVIDIA, 2023. URL: https://research.nvidia.com/labs/rtr/neural_appearance_models/ 1.
- [ZSD*21] ZHANG, XIUMING, SRINIVASAN, PRATUL P., DENG, BOYANG, DEBEVEC, PAUL, FREEMAN, WILLIAM T., and BARRON, JONATHAN 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](https://doi.org/10.1145/3478513.3480496) 2, 7, 9.