Eurographics Symposium on Rendering 2024
E Garces and E. Haines
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 4

Neural SSS: Lightweight Object Appearance Representation

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

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

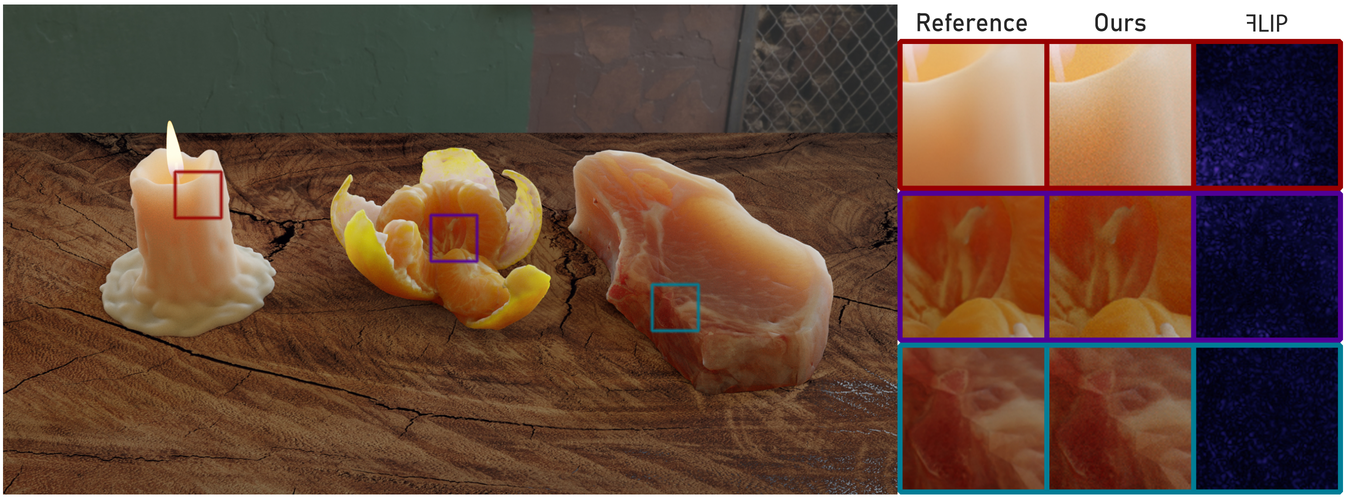

Figure 1: *Three translucent objects each represented in only 0.77 MB using our neural subsurface scattering (SSS) representation. The candle is a homogeneous medium while the mandarin and the meat are heterogeneous in nature. The flame above the candle locally illuminates the objects while the environment is distant. The meat is a scanned object whereas the mandarin is completely synthetic. This scene illustrates our model's ability to capture the full 8-dimensional BSSRDF of an object and represent it faithfully as demonstrated in the* ꟻLIP *error maps.*

Abstract

We present a method for capturing the BSSRDF (bidirectional scattering-surface reflectance distribution function) of arbitrary geometry with a neural network. We demonstrate how a compact neural network can represent the full 8-dimensional light transport within an object including heterogeneous scattering. We develop an efficient rendering method using importance sampling that is able to render complex translucent objects under arbitrary lighting. Our method can also leverage the common planar half-space assumption, which allows it to represent one BSSRDF model that can be used across a variety of geometries. Our results demonstrate that we can render heterogeneous translucent objects under arbitrary lighting and obtain results that match the reference rendered using volumetric path tracing.

CCS Concepts

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

1. Introduction

Subsurface scattering is a captivating visual phenomenon characterised by its unique soft look. It plays a pivotal role in the appearance of a wide range of materials such as skin, fruits, minerals, etc. However, physical simulation of this phenomenon using Monte Carlo path tracing has always been computationally expensive due to long chains of scattering events that occur behind a refractive interface until the path exits the medium. Many alternative analytic formulations based on diffusion theory have been introduced as global formulations of subsurface scattering. Although these models are more practical than physical Monte Carlo simulations, they introduce error since diffusion theory derives its formulation by imposing several assumptions such as a planar surface, homogeneity, and isotropy that are more often than not violated in the real world. These reasons lead rendering systems to prefer volume path tracing despite its computational expense.

We introduce a neural subsurface scattering (SSS) method, a compact learned volumetric light transport model based on Monte Carlo simulations that can represent the full light transport of an object exhibiting translucency (see examples in Figure 1). We train our model on standard path tracing data that many existing rendering frameworks can generate.

As is commonly done, we distinguish between surface and subsurface scattering effects and evaluate them separately. While subsurface scattering effects are complex to model due to long chains of Monte Carlo scattering in a path tracer, surface reflections are local and oftentimes of high frequency. We use ray tracing and a standard bidirectional scattering distribution function (BSDF) to capture surface scattering and a multi-layered perceptron (MLP) as our base for representing the subsurface scattering.

One way to formulate the subsurface scattering is as a sampled estimate of the volumetric light transport. Having N samples per iteration in a progressive rendering allows for better convergence at lower sample counts and eliminates the need for commonly used feature extraction [MST*20, TSM*20, TFRJ23] on low dimensional inputs. Our compact neural network architecture enables our training to be conducted in parallel to an active renderer that generates data. This approach saves time and resources by avoiding explicit training after data generation. Despite the BSSRDF being 8-dimensional in nature (two surface positions and two directions) and being potentially different for different wavelengths or color bands, we find it advantageous to let our network learn a function of $5+N\times 8$ dimensions, where the first five dimensions denote the position and direction of observation ($\boldsymbol{x}_o$ as a 3D point and $\vec{\omega}_o$ as a unit length direction vector in spherical coordinates). The remaining $N\times 8$ dimensions constitute N samples of positions and directions of incidence ($\boldsymbol{x}_i$ and $\vec{\omega}_i$) together with an RGB vector of incident radiance ($L_i(\boldsymbol{x}_i,\vec{\omega}_i)$).

Our key finding is that we do not need converged rendered observations of surface points to represent the object appearance. Each instance of our training data is simply the result of N paths traced from one position and direction of observation ($\boldsymbol{x}_o,\vec{\omega}_o$) to N positions and directions of incidence ($\boldsymbol{x}_{i,j},\vec{\omega}_{i,j},\ j=1,\ldots,N$) reached by randomly sampled paths through the volume from this outset and the contribution to the observation of the radiance incident in these places ($L_{i,j}$). When the network has seen sufficiently many examples of such input, it is surprisingly good at performing an N-samples volumetric path tracing for us. Due to the ability of neural networks to smoothly interpolate between different samples seen during training, the network output for N samples is no longer as when tracing N random paths but more like the BSSRDF result for the N samples. Thus, we get less noise during progressive rendering and fast convergence to a surprisingly accurate result with our N-samples subsurface light transport network.

Our model is in the outset intended for representation of the appearance of a specific object of known surface geometry and heterogeneous optical properties. After learning the light transport between surface positions and directions, we still need the surface geometry to find the appropriate input positions and directions for the model when rendering the object. To include a shape-adaptive model similar to existing analytic BSSRDF models, we include an option to formulate our neural subsurface scattering model on a planar half-space to make it a function of distance and directions instead of positions on a given 3D surface. Since our model is a learned approximation of a surface-to-surface transport function with full directionality and no assumption of high scattering, our method attains better accuracy than analytic models.

For practical use of our model in a rendering, we introduce a novel object-specific importance sampling technique for sampling a collection of surface positions, where incident light should be sampled, based on a normalizing flow architecture. This method enables us to map simple distributions such as Gaussians to a more complex distribution by attaining an invertible function that allows exact posterior inference.

2. Related Work

2.1. Monte Carlo Ray Tracing

The reference technique for rendering of a translucent object is volume path tracing [NGHJ18], which involves a random walk within the medium that proceeds until it is stopped by absorption or emerges from the medium. This quickly becomes inefficient as many samples have no contribution. Several techniques have been presented to mitigate this issue: bidirectional volume path tracing [LW96], volume photon mapping [JC98], metropolis light transport [PKK00], bidirectional lightcuts [WKB12], vertex connection and merging [KGH*14], residual ratio tracking [NSJ14], and path guiding [MHD16, HZE*19, DWWH20]. These methods however still do the random walk within the medium.

To accelerate the random walk, Kallweit et al. [KMM*17] introduced Radiance-Predicting Neural Networks (RPNNs), where the multiple scattering solution is learned for a volume density descriptor in the form of a hierarchical point stencil oriented in the direction toward a distant source. This technique has been further accelerated with interpolation of light probes precomputed for a distant directional light as input for the RPNN [PN19] and adapted to volumes other than clouds [RSB*21]. The original RPNN technique [KMM*17, PN19] still does the random walk inside to capture single scattering and the assumption of a distant directional light makes the method unsuitable for a local lighting environment. In addition, it is unclear how well the density descriptors would work for a volume with a refractive interface, as the distribution of light would also depend on the surface shape. In the adaptation of the technique for predicting the color of 3D printed objects [RSB*21], the assumption of directional incident light is replaced with an assumption of diffuse incident light and observation along the surface normal. The RPNN is thus modified to capture a non-directional BSSRDF stencil, but the missing directionality makes the method unsuitable for arbitrary relighting of the object.

The original BSSRDF captures all the subsurface scattering with full directionality and thus enables us to consider only surface positions when rendering an object under arbitrary view and lighting configurations. The only assumption is that no light sources are placed within the medium. This usually leads to rendering techniques that are orders of magnitude faster than volumetric path tracing. The difficulty is to accurately represent the BSSRDF as it in principle depends on the full object surface geometry and the variations of the optical properties inside the medium while it is also a function of the positions and directions of incidence and observation. Our intention is to provide a neural representation that captures this complexity without the many simplifying assumptions used for analytic BSSRDF models.

2.2. Diffusion-Based Models

Based on diffusion theory, the light transport in a translucent object can be reformulated as a partial differential equation (PDE) problem solvable by finite element methods [Sta95, HMBVR05, WZT*08, WWH*10, AWB11]. Diffusion theory however involves simplifying assumptions such as high scattering, low absorption, and scattering taking place far from sources and boundaries. The latter is obviously violated in practice, leading to inaccuracies in the rendered result. Further simplification can lead to fully analytic BSSRDF models [JMLH01, DJ05, dI11, FHK14], which further assume a planar half-space medium or layer. Boundary conditions are difficult to model accurately in diffusion theory, so diffusion-based models oftentimes also assume no dependence on the directions of incidence and observation (or just observation [FHK14]). Extended source models mitigate this problem by integrating an analytic solution along a refracted ray of incident light [DJ07, YZXW12, HCJ13], but this also misses dependence on the direction of observation. An alternative model includes full directionality of the incident and emergent light by assuming a very forward-scattering medium [FD17]. This model is not based on diffusion but derived from a functional integral approximation of radiative transfer, which means that an additional integration is required over all path lengths within the medium. By assuming a half-space homogeneous translucent object, our method can be used in the same way as the analytic models but without assuming high scattering, low absorption, or limited directionality of the incident, scattered, and emergent light. In addition, by assuming fixed surface geometry, we can represent an arbitrary heterogeneous translucent object without the simplifying assumptions of diffusion theory.

2.3. Texture-Based Approximations

Texture-based approaches are useful for representing spatial variation of optical properties across a surface. Using these spatially varying optical properties as input for an analytic BSSRDF model, one can model the appearance of heterogeneous materials [GLL*04, PvBM*06, CPZT12, KÖP13, Kur21, DLW*22]. Surface variation of optical properties, however, can only roughly approximate the 3D variation of the properties in a heterogeneous translucent object. A bidirectional texture function (BTF) [DVGNK99, RGJW20, KMX*21, KWM*22, FWH*23] is a bidirectional reflectance distribution function (BRDF) spatially extended to a flat textured patch of surface geometry. As opposed to a spatially varying BRDF, which assumes that light is incident and observed in the same surface position, a BTF can include subsurface scattering within the patch. However, the BTF assumes light incident from one direction across a flat surface and is thus best suited for distant lighting and scenarios where the bleeding of light through the object is not important. Combination with an analytic model is an option to partially mitigate the latter problem [TWL*05]. The assumptions of distant lighting and scattering within a patch make the BTF models unable to fully capture the appearance of a heterogeneous translucent object. Our model enables this by considering the full object geometry as in precomputation-based models described next.

2.4. Precomputation

Different representation techniques enable precomputation of the light transport within a translucent object. An option is to discretize the object surface into polygons and compute the transport between them due to subsurface light transport [LGB*02, SSWN14]. These methods however tend to be limited by the size of the transfer matrix needed to define the polygon-to-polygon transport. Additionally, the models assume diffuse emergent light to reduce the information that needs to be stored. To include full directionality, subsurface scattering can be precomputed across a planar surface [DLR*09] or in a sagittal and a coronal plane for a full object [WGH19] and stored in a multidimensional array for look-up. These approaches, however, do not generalize well to heterogeneous translucency. Precomputed radiance transfer (PRT) methods represent the subsurface light transport in zonal harmonics [SLS05], wavelets [WTL05], transfer functions [BDBS22], or a neural network [TFRJ23]. While such representation can achieve real-time rendering at run time, they are in all these cases limited to an assumption of distant lighting. Our model does not suffer from these limiting assumptions.

2.5. Neural Models

A neural BSSRDF was first presented by Vicini et al. [VKJ19]. They avoid the assumption of a flat surface by including an implicit functional representation of the local geometry as input for their model. However, as opposed to our work, they assume a homogeneous material and a diffuse distribution of the incident light. The shape-adaptive version of our model could be improved by an extension of this kind to represent variation of local geometry. Neural radiance fields (NeRF) [MST*20] is a method based on ray marching useful for representing object appearance in a fixed lighting environment. To be useful in the context that we consider, the NeRF must be relightable. Most relightable NeRF models however rely on an assumption of distant lighting (as for PRT) [SLB*21, LGF*22, YGF*23, ZSB*23] or a spatially varying BRDF [BBJ*21, ZSD*21, LTL*22, ZSH*22, YZL*22]. Those assuming a BRDF are unable to capture translucency effects, as these are generally a consequence of the position of observation being different from the position where light is incident. Some relightable neural models include subsurface scattering and an option for local illumination due to being trained with point light illumination [GCD*20, ZCD*23]. While being able to use captured images to learn the appearance of physical objects, these models include surface scattering and struggle with highlights and surface reflections. Since we work with digital objects, we can easily separate out these effects.

3. Method

Our objective is to represent the subsurface light transport of an object. Like a BSSRDF, the representation should care only about incident and emergent light at the object surface, but volume path tracing is used to compute the subsurface scattering. After refraction through the surface of an object, the transport of light is described by the radiative transfer equation [Pre57]:

$$n(s)^2 \frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{L(s)}{n(s)^2}\right) = -\sigma_t(s)\,(L(s)+J(s))\,, \tag{1}$$

where n is the refractive index of the material, s is distance traveled along a ray defined by $\boldsymbol{x}(s) = \boldsymbol{x}_o + s\,\vec{\omega}_o$, $L(s)$ is the radiance at $\boldsymbol{x}(s)$ in the direction of the ray $\vec{\omega}_o$, $\sigma_t = \sigma_a + \sigma_s$ is the extinction coefficient (absorption and out-scattering), and J is the source function (which includes in-scattering) defined by

$$J(s) = \alpha(s)\int_{4\pi} p(s,\vec{\omega}_i,\vec{\omega}_o)L(s,\vec{\omega}_i)\,\mathrm{d}\omega_i + \varepsilon(s)\,, \tag{2}$$

where ε is emission (only non-zero for emissive materials) and $\alpha = \sigma_s/\sigma_t$ is the scattering albedo. At a point in the medium, α is the probability of scattering while $1-\alpha$ is the probability of absorption. We note that in our case, where the path starts and ends in the same medium surrounding an object, the squared refractive index (n^2) cancels out of Eq. 1.

The integral form of the radiative transfer equation (with n^2 omitted) is sometimes referred to as the volume rendering equation [NGHJ18]. This is [Cha50]

$$L(s) = L(0)e^{-\tau(0,s)} + \int_0^s \sigma_t(s')J(s')e^{-\tau(s',s)}\,\mathrm{d}s'\,, \tag{3}$$

where $\tau(s',s) = \int_{s'}^{s}\sigma_t(t)\,\mathrm{d}t$ is the optical thickness of the material and $L(0)$ is light entering the medium at the surface. Volume path tracing solves the integral by Monte Carlo integration. Starting at the distance s along the ray, since we trace from the observer, the trick is to sample the distance to the next scattering event $s-s'$ using the probability density function

$$\mathrm{pdf}(s') = \sigma_t(s')e^{-\tau(s',s)}\,. \tag{4}$$

The probability of getting $s' \geq s$ corresponds to a Russian roulette on whether to include the first term of Eq. 3 or not. The Monte Carlo estimator with this sampling scheme is then

$$L_N(s) = \frac{1}{N}\sum_{j=1}^{N}\begin{cases} J_M(s'_j) & \text{for } s'_j < s \\ L(0) & \text{for } s'_j \geq s, \end{cases} \tag{5}$$

where J_M is an M-samples Monte Carlo estimator for Eq. 2 in which we always use $M = 1$.

One obstacle is that the optical properties depend on wavelength and the path will then be different for different color bands. To solve this problem, we uniformly sample a color band for each path that we trace. We set the contribution of the other color bands to zero for this path and divide the result by the probability of the sampled band (1/3 for uniform sampling). Another problem is that we have no control over the positions and directions of incidence reached by the paths we trace. To evaluate the subsurface scattering for arbitrary positions and directions of incidence, we need the BSSRDF. Fortunately, one way to define the BSSRDF is using an infinite collection of paths traced by volume path tracing. Based on a large enough set of samples, a neural network can be trained to represent such an infinite collection of paths. In the following, we use this to justify the use of a learned N-samples Monte Carlo subsurface scattering function in a surface-based rendering technique that allows us to sample surface positions and directions without performing a random walk through the object interior.

In Sec. 3.1, we present our N-samples light transport function that captures the subsurface-scattering effects from the full geometry of a heterogeneous translucent object. In Sec. 3.2, we describe how we can also represent the light transport function of a planar half-space. Violating the assumption of a planar half-space (as is commonly done with analytic BSSRDFs), we can use this model with arbitrary surface geometry. Finally, we describe importance sampling of our light transport functions in Sec. 3.3.

3.1. Full Geometry

Suppose X is a closed object defined by its surface geometry G and its surface and volume scattering properties. The radiance L_r reflected towards an observer from the point $\boldsymbol{x}_o \in G$ in the direction $\vec{\omega}_o$ is a function of all the light incident across the object surface G. If the point of incidence is $\boldsymbol{x}_i \in G$ and the direction of incidence is $\vec{\omega}_i$, the BSSRDF is a function describing the part of the incident flux Φ_i that contributes to L_r. The definition of the BSSRDF is [Pre65, FJM*20]:

$$S(X;\boldsymbol{x}_i,\vec{\omega}_i;\boldsymbol{x}_o,\vec{\omega}_o) = \frac{\mathrm{d}L_r(\boldsymbol{x}_o,\vec{\omega}_o)}{\mathrm{d}\Phi_i(\boldsymbol{x}_i,\vec{\omega}_i)}\,. \tag{6}$$

Using the definition of radiance [Nic63], we get the reflected radiance equation:

$$L_r(\boldsymbol{x}_o,\vec{\omega}_o) = \int_G\int_{2\pi} S(X;\boldsymbol{x}_i,\vec{\omega}_i;\boldsymbol{x}_o,\vec{\omega}_o)L_i(\boldsymbol{x}_i,\vec{\omega}_i)\cos\theta_i\,\mathrm{d}\omega_i\,\mathrm{d}A_i\,, \tag{7}$$

where L_i is incident radiance, 2π denotes the solid angle of the hemisphere centered around the surface normal $\vec{n}_i$ at the location $\boldsymbol{x}_i$ and $\cos\theta_i = \vec{\omega}_i\cdot\vec{n}_i$. In the integration, $\mathrm{d}\omega_i$ is a differential solid angle around $\vec{\omega}_i$ and $\mathrm{d}A_i$ is a differential area around $\boldsymbol{x}_i$.

Rendering with a BSSRDF requires integrating it across the object surface (as seen in Eq. 7). With a neural representation, this becomes a very expensive operation, which is likely one of the reasons why current neural techniques [GCD*20, ZCD*23, TFRJ23] do not represent the full function. Instead, they assume distant lighting or point lighting to include the integral across the surface area in the model. We model the full function without sacrificing efficient rendering by providing our model with a distribution of surface points each with associated direction of incidence and incident radiance, that is, a collection of $(\boldsymbol{x}_i,\vec{\omega}_i,L_i)$ triplets.

For any $(\boldsymbol{x}_o,\vec{\omega}_o)$ pair, we can perform a surface scattering event. If the result is refraction into the medium, we use unidirectional volume path tracing in the direction of the refracted ray without a probabilistic stopping criterion to connect these BSSRDF arguments with a position and a direction of incidence $(\boldsymbol{x}_i,\vec{\omega}_i)$ where the path emerges from the medium. Accounting for all the paths, we can define a version of the S-function that takes a set of points of incidence G_i and a set of directions of incidence Ω_i as argument instead of a particular position and direction of incidence [Pre65]:

$$S(X;G_i,\Omega_i;\boldsymbol{x}_o,\vec{\omega}_o) = \frac{\mathbf{S}[L_i](\boldsymbol{x}_o,\vec{\omega}_o)}{L_i(G_i,\Omega_i)\,A_i(G_i)\,\omega_i(\Omega_i)}\,. \tag{8}$$

The sets G_i and Ω_i are infinite sets of positions and directions forming an illuminated part of the object surface and a solid angle in which light is incident. The term $L_i(G_i,\Omega_i)$ is the sum of radiances incident in G_i and Ω_i, $A_i(G_i)$ is the area of the set of points G_i (corresponding to projected area in definitions with differential quantities), and ω_i is the solid angle of the set of directions Ω_i, while $\mathbf{S}[L_i]$ is a scattering operator applied to L_i denoting the sum of radiance due to all paths from some starting point in G_i and Ω_i reaching the position and direction of observation (Eq. 3). The conventional BSSRDF is the limit of this function for $G_i \to \boldsymbol{x}_i$ and $\Omega_i \to \vec{\omega}_i$.

There is nothing preventing us from taking multiple disjoint limits such that $G_i \to \{\boldsymbol{x}_{i,1},\dots,\boldsymbol{x}_{i,N}\}$ and $\Omega_i \to \{\vec{\omega}_{i,1},\dots,\vec{\omega}_{i,N}\}$. The S-function then becomes a matrix function $\boldsymbol{S}_N$ with an output in $\mathbb{R}^{N\times 3}$ (where 3 is for RGB). Each RGB row of the matrix is obtained by a volume path tracing corresponding to evaluation of Eq. 5 divided by the incident flux. Using this notation for matrices containing a discrete set of N samples, we write Eq. 7 in an N-samples version:

$$L_r(\boldsymbol{x}_o,\vec{\omega}_o) = \int_G\int_{4\pi} (\boldsymbol{S}_N \otimes \boldsymbol{L}_{i,N})^\top \mathrm{diag}(\boldsymbol{n}_{i,N}\,\boldsymbol{\omega}_{i,N}^\top)\,\mathrm{d}\omega_i\,\mathrm{d}A_i\,, \tag{9}$$

where $\otimes$ is elementwise mutliplication while $\boldsymbol{L}_{i,N}$, $\boldsymbol{n}_{i,N}$, and $\boldsymbol{\omega}_{i,N}$ are N-samples matrix versions in $\mathbb{R}^{N\times 3}$ of the incident radiance, the surface normals, and the directions of incidence at the respective positions of incidence. This is the equation we evaluate during a rendering, and $(\boldsymbol{S}_N \otimes \boldsymbol{L}_{i,N})^\top \mathbf{1}$ is the function we let our neural network represent, where $\mathbf{1} = \mathrm{diag}(I)$ is the diagonal of the identity matrix.

Creating a multi-layer perceptron (MLP) f_ψ, with parameters ψ, which takes one $(\boldsymbol{x}_o,\vec{\omega}_o)$ pair and triplets $(\boldsymbol{x}_{i,j},\vec{\omega}_{i,j},L_{i,j})$ for $j = 1,\dots,N$ as input and returns an RGB vector, we find parameters $\psi*$ representing the appearance of a given object by

$$\psi* = \arg\min_\psi \mathbb{E}\left[\left(f_\psi - (\boldsymbol{S}_N\otimes\boldsymbol{L}_{i,N})^\top\mathbf{1}\right)^2\right]\,, \tag{10}$$

To estimate $(\boldsymbol{S}_N \otimes \boldsymbol{L}_{i,N})^\top \mathbf{1}$, we use the volume path tracing mentioned for its definition. When a path emerges from the medium, the path would usually continue to evaluate incident radiance. For training of the MLP, we use a random incident radiance $L_{i,j}$. This path tracing corresponds to evaluation of Eq. 5 with random $L(0)$. Because of the randomization during training and because the incident radiances are also input to the network, the model becomes fully relightable when used for rendering, where we evaluate Eq. 9 by Monte Carlo integration. To use the network in this Monte Carlo integration, we set $L_{i,j}$ in the N triplets given as input to the network to sampled cosine-weighted incident radiances divided by their respective sample probabilities.

3.2. Planar Half-Space

For the case of an object X that is a planar half-space with a homogeneous material, we simplify our model and use $N = 1$. The rest is the same as before. The difference is illustrated in Figure 2.

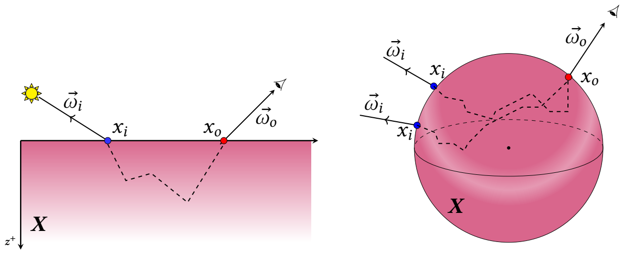

Figure 2: *The difference between a half-space model (left) and the full geometry version of our method (right). Our model implicitly depends on the full object geometry X whereas analytic BSSRDF methods assume a planar half-space geometry. Our method can adapt to a planar half-space representation and then no longer depends on the object geometry.*

The planar half-space version of our method provides a fully directional BSSRDF function that we can apply to arbitrary geometry in the same way as analytic BSSRDF models are used.

This geometry simplifies the problem to fewer dimensions as compared with the full object BSSRDF, as the coordinates are in 2D space and we only need to consider $\boldsymbol{x}_o - \boldsymbol{x}_i$. We thus have only six dimensions (two for each spherical direction and two for the spatial difference vector). To use this model for arbitrary geometry, we either project the point of incidence to a representative plane or use only the distance between the points of incidence and observation $\|\boldsymbol{x}_o - \boldsymbol{x}_i\|$ together with the directions of incidence and observation as inputs for our network representation.

3.3. Importance Sampling

To efficiently evaluate Eq. 9 by Monte Carlo integration, we propose an importance sampling method based on normalizing flow [DKB15, MMR*19]. Conventional uniform sampling of the object surface and sampling of a cosine-weighted hemisphere for directions of incidence can be used with our neural representation. However, for local illumination as in the candle scene in Figure 3, a uniform sampling would generate many position-direction pairs without direct visibility of the light source. Importance sampling techniques for directions have been studied extensively and most of them are compatible with our method. Thus, we find it crucial to have a robust scheme for importance sampling points of incidence $\boldsymbol{x}_i$ that can work alongside any direction importance sampling method [MMR*19, XWH*23].

An importance distribution is easily generated from observed data. Efficiently sampling such a distribution is however a challenge. Given a difficult to sample probability distribution p_Z and a distribution p_Y that we can efficiently sample, if we know a bijective transformation $f: Z \to Y$ then sampling $\mathbf{z} \in Z$ can be achieved using $\mathbf{z} = f^{-1}(\mathbf{y})$. We use a normalizing flow technique to find the function f. A normalizing flow has the property of being diffeomorphic, which means that any transformation f in the flow is invertible and both f and f^{-1} are differentiable. This also necessitates that the input dimensions of the distribution match the output dimensions. We use a 3D Gaussian as our base distribution for ease of sampling and density evaluation.

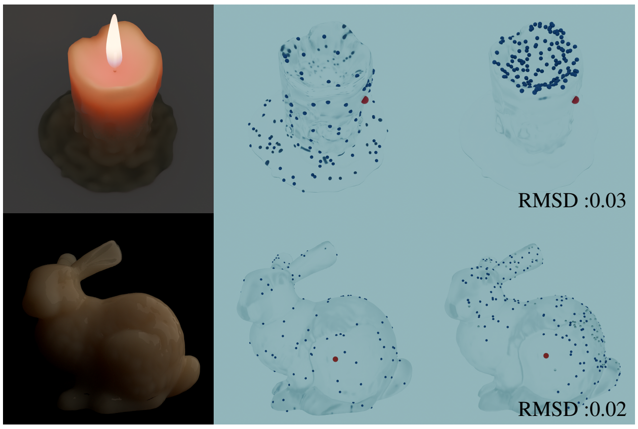

Figure 3: *Examples of our importance sampling method applied to a locally lit object (top) and an object lit by a distant source (bottom). The second column of images shows positions sampled uniformly on the surface. The third column shows a distribution of importance sampled positions $\boldsymbol{x}_{i,j}$ (blue) for a given $\boldsymbol{x}_o$ (red). The faded blue points represent the far side of the mesh. In the candle scene, our method learns that the important region is the concave part with direct visibility of the flame. In the bunny scene, a broader distribution of points is sampled where the scene is illuminated by the directional light. Note that our method is not constrained to sampling points strictly on the mesh surface. As an upper bound on the distance from the sampled points to the surface, we list the root-mean-squared deviation (RMSD) for the sampled points with respect to the closest barycenter.*

The loss function for the maximum likelihood objective is [DKB15]

$$\log p_Y(\mathbf{y}) = \log p_Z(\mathbf{z}) + \log\left|\det\left(\frac{\partial \mathbf{z}}{\partial \mathbf{y}}\right)\right|, \tag{11}$$

which preserves the volume under the curve while providing exact posterior inference (via $z = f^{-1}(y)$). This loss function is used for training the 2-layers MLP in our importance sampling architecture sketched in Figure 4.

Our network architecture takes N positions x_i and directions ω_i as input. We take advantage of the normalizing flow architecture by generating a set of points representing the base distribution. Thus, a single f^{-1} can be used to attain the samples in the real distribution (see Figure 4). This creates a collection of good samples instead of conducting individual inference for each sample.

4. Implementation

Our neural network has two modules, an importance sampling module and an appearance representation module. We use a Multi Layered Perceptron (MLP) to represent $(\boldsymbol{S}_N \otimes \boldsymbol{L}_{i,N})^\top \mathbf{1}$. A tiny MLP is storage efficient as compared to nearly all precomputation alternatives [LGB*02, SSWN14, BDBS22].

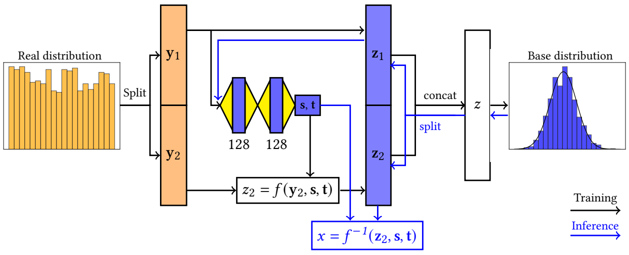

Figure 4: *Our normalizing flow architecture for importance sampling. We collect a set of samples from the real distribution based on the output from our neural network. This distribution goes through a set of diffeomorphic transformations that are invertible allowing for exact posterior inference. We use a 3D Gaussian as a base distribution. During inference, we resample the 3D Gaussian for every ray hit on the object and go through the inverse of the entire transformation which returns a vector with N sampled 3D positions on the object geometry. The black line in the graph is a training loop whereas the blue lines represent the inference process.*

4.1. Network Architecture

Object appearance. MLPs have demonstrated their usefulness with respect to reliable representation of high-dimensional functions. As our material representation is a function of $5 + 8 \times N$ dimensions, an MLP should serve our purposes well. We use a standard 3-layers fully connected MLP with 256 nodes in each layer to represent object appearance both for full geometry and a planar half-space. We use a LeakyReLU activation function after each layer. Furthermore, we use the mean-squared error (MSE) as the loss function ($\mathcal{L}$) for our MLP (Eq. 10). A bigger network should in principle be able to learn the light transport function better, but this simple architecture provided a suitable fit in all our test cases, and we therefore use it throughout. We described two different networks in Sec. 3: an object-centric full geometry representation and a half-space representation useful for arbitrary geometries. We use the same network for the half-space formulation but with $N = 1$.

Importance sampling. As described in Sec. 3.3, we use a normalizing flow architecture with a coupling layer for our importance sampling module. We first create a small dataset using our appearance specification module by uniformly sampling points on the object surface G. For each point, we integrate out the dependence on the direction of incidence. In practice, we use 16×64 uniformly distributed direction vectors for this integration. A number of evaluations of the network are then converted into a distribution, such that

$$p_Y(\mathbf{y}) = \frac{\int_{4\pi} (\boldsymbol{S}_{N,k} \otimes \boldsymbol{L}_{i,N,k})^\top \operatorname{diag}(\boldsymbol{n}_{i,N,k}\, \boldsymbol{\omega}_{i,N,k}^\top)\, \mathrm{d}\omega_i}{\sum_k \int_{4\pi} (\boldsymbol{S}_{N,k} \otimes \boldsymbol{L}_{i,N,k})^\top \operatorname{diag}(\boldsymbol{n}_{i,N,k}\, \boldsymbol{\omega}_{i,N,k}^\top)\, \mathrm{d}\omega_i}, \tag{12}$$

where k is the index of a data point in the set, and each data point consists of N surface samples with the direction of incidence integrated out. The created distribution goes through a set of transformations such as splitting the data, a coupling transform and finally, the concatenation of the transformations in the end, as depicted in Figure 4. We do the data splitting in two equal subsets $\mathbf{y}_1$ and $\mathbf{y}_2$.

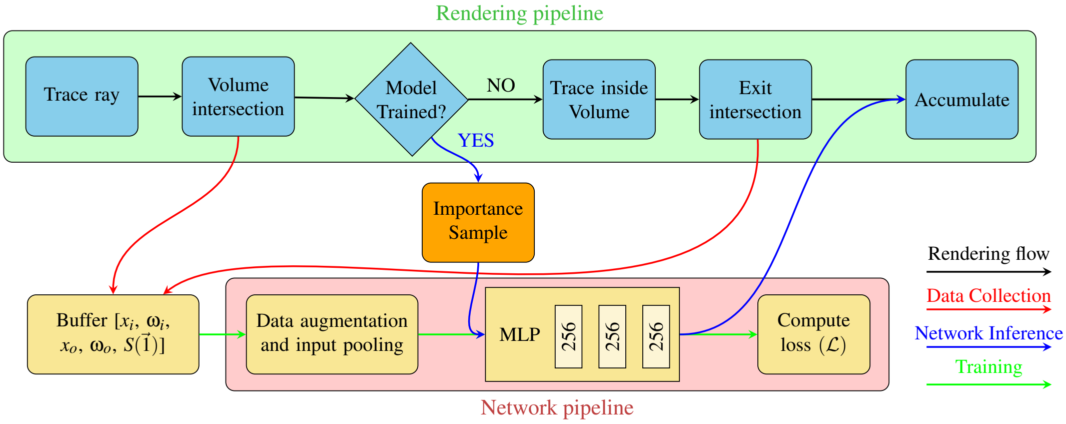

Figure 5: *A process flow diagram depicting our method plugged into a standard volume path tracer. The "Model Trained" variable is initially "NO", and the path tracer then stores the intersection and radiance values in the training buffer. The data augmentation block takes data from the buffer, pools it as inputs and trains the network. We set the "Model Trained" parameter to "YES" when the predicted loss is less than* 10^{-3} *for* 100 *iterations in a row. When the model has been trained, we do importance sampling for the exit positions and sample the incident radiance at each point. This is fed into the network which in turn predicts the radiance output used in the accumulation step.*

We use an identity transform on $\mathbf{y}_1$ whereas $\mathbf{y}_2$ serves as input to a 2-layers MLP with 128 neurons each and ReLU activation in-between to provide values for the scale $\mathbf{s}$ and the transpose $\mathbf{t}$. The scale and the transpose are used for a linear polynomial transformation of the subset $\mathbf{y}_2$, such that

$$f(\mathbf{y}_2) = \mathbf{y}_2 \otimes \mathbf{s} + \mathbf{t}, \tag{13}$$

where the transformed subset is concatenated with $\mathbf{z}_1$. The set is shuffled and this iteration continues until a satisfactory fit is obtained. Our importance sampling technique is a one-to-one mapping from a 3D Gaussian to the real distribution. However, this only ensures that the sampled points are close to the surface and does not necessarily ensure that they are on the surface. A point to mesh projection technique is therefore beneficial. We project our points on the surface by sampling a random point on the closest triangle.

4.2. Training and Dataset

We train our network on data produced by an unbiased path tracer with an arbitrary scene containing the object of interest. Our network has a tiny memory footprint (0.77 MB), which allows us to train while rendering any scene with negligible cost. Thus, we employ an online learning method to train our appearance specification network, see Figure 5. The online learning attains quicker model creation by parallelizing data generation and training.

Most online learning methods impose restrictions on the data generation process. This limits the usability of the generated data. Our online learning makes use of standard path-tracing data produced by a renderer without imposing any such restrictions. This makes the online learning process more functional by saving explicit time consumed by training after data collection in an offline method. The training data is generated by path tracing N times through the object for every trace hit on the geometry. We keep track of all the trace exit point-direction pairs $(\boldsymbol{x}_i, \vec{\omega}_i)$ on the geometry while the ray hit position and the negated ray direction become our $\boldsymbol{x}_o$ and $\vec{\omega}_o$. We generate additional training data by scaling the generated transport information ($\boldsymbol{L}_{i,N}$ and the network output) which is analogous to "data augmentation" in classic machine learning. We train the model until the MSE error from the network prediction is below 10^{-3} for 100 consecutive iterations.

Our *planar half-space* approach has a different method of data collection where we trace a beam of light with unit radiance through a planar half-space. Unlike the other N-samples formulation, this method is a single sample representation making it low-dimensional. For training this formulation, we trace paths through a planar half-space from a single point $\boldsymbol{x}_i$ at evenly distributed 16×64 spherical coordinates. Every incident ray creates a 2D map of data consisting of $\boldsymbol{x}_o$, $\vec{\omega}_o$ and exit radiance L_r. This map is used to train our network for every incident ray. We fix $\boldsymbol{x}_i$ to be the origin of a 2D Cartesian coordinate system and the position of observation $\boldsymbol{x}_o$ would be relative to the origin. Each pixel in a path traced 2D map is a data point for the network i.e., $\boldsymbol{x}_i$, $\vec{\omega}_i$, $\boldsymbol{x}_o$, $\vec{\omega}_o$. We train this formulation offline where we collect path tracing data and train on it independently. Once this model is trained for a medium, the same model can be used for rendering arbitrary geometries. This model is however less accurate than full object representation as demonstrated in Figure 9.

4.3. Framework and Rendering

For efficient data collection and rendering, we use a custom-made path tracing framework based on NVIDIA OptiX [PBD*10]. The neural network undergoes training on a C++ distribution of PyTorch, commonly referred to as libTorch [PGM*19]. Network inference is performed on a custom implementation of MLP for the OptiX framework. All tests were performed on an NVIDIA RTX 4090 and an AMD Ryzen 7 5800X3D.

We implemented our models as shader code in our Monte Carlo

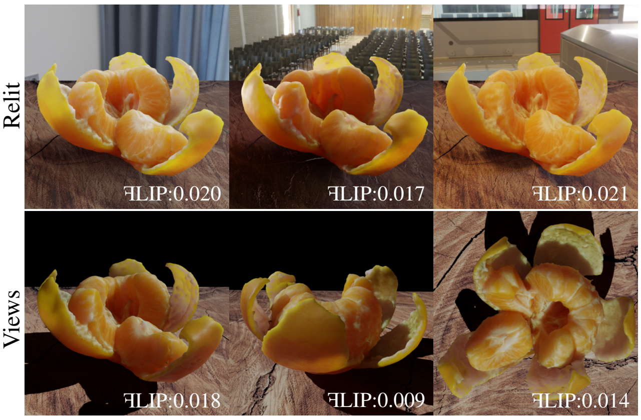

Figure 6: *Example renderings to demonstrate that our model supports relighting and an arbitrary view. We use the same representation for all the figures. The first row of images is rendered a variety of environment maps showing that the views can be rendered under a variety of lighting setups. The second row of images are rendered under the same directional source with a variety of views.*

path tracer where we resample the 3D Gaussian which is translated into $\boldsymbol{x}_i$ positions, and we sampled $\vec{\omega}_i$ on a cosine-weighted hemisphere or directly toward light sources. This information along with $\boldsymbol{x}_o$ and ω_o is passed through one of our two neural subsurface scattering representations which return a radiance output (3-vector in case of RGB). Note that since our planar half-space representation is trained without data augmentation using unit incident radiance and one position of incidence, we directly scale the network output (instead of the input radiance) by the sampled radiance from $\vec{\omega}_i$ at $\boldsymbol{x}_i$. As the planar half-space formulation inherently disregards geometry as a whole, we use the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_o$ for the model when rendering 3D scenes.

5. Results

Figure 6 is a rendering of the same model under different lighting conditions and viewpoints. We use average ꟻLIP error [ANAM*20] to assess the perceptual difference between our result and that of a reference path traced image. The low values demonstrate that our model achieves high fidelity while the object appearance representation is fully relightable and view-independent.

To demonstrate that our method accurately represents the appearance of translucent objects, we compare our method with path traced reference images. To illustrate the benefits of accounting for the full object geometry, we also compare with related work based on conventional analytic BSSRDF models [DLW*22]. In addition, we compare with a neural object-centric appearance representation technique that assumes distant lighting [TFRJ23]. Finally, we demonstrate the effectiveness of our importance sampling method.

5.1. Comparison with Other Methods

Figure 1 shows three translucent objects rendered under environment lighting and with local illumination from the flame in the candle model. We compare against the reference path traced image. Again, the low ꟻLIP error attests to the high fidelity of our model. Figure 7 is a comparison between the reference path tracing, the methods of Deng et al. [DLW*22] and TG et al. [TFRJ23], and our method.

Table 1: *Training and rendering times in seconds for our models in Figures 7 and 9, and the number of samples that we used when rendering with our method. Note that the planar half-space formulation is an offline training method which takes approximately equal training and inference time for every material. However, the data generation time differs from material to material. Every model consumes 0.77 MB of storage space.*

Model	Training time	Render time	Samples
Mandarin	1513.12	36.13	2200
Meat	1392.12	32.16	2100
Candle	1312.13	18.27	1500
Croissant	1657.78	27.16	1300
Planar half-space	2118.12	83.58	5000

Deng et al. [DLW*22] introduced an inverse rendering technique to jointly optimize spatially distributed BSSRDF parameters for the standard dipole [JMLH01] using a set of images. This method achieves a good result for a highly scattering medium such as the croissant, however, the underlying assumption of the standard dipole model, which assumes a homogeneous half-space material, is not a good representation of the thin heterogeneous objects seen in the mandarin and meat scenes. Interestingly, this model also has difficulties with the candle scene although the material is homogeneous. This is likely due to some of the other limitations of the standard dipole model, such as the assumption of no directional dependence of the diffuse part of the subsurface scattering.

TG et al. [TFRJ23] on the other hand achieve a good overall representation, however, their model cannot reliably represent local illumination effects due to their assumption of distant lighting. This is easily seen in the candle scene where the flame does not accurately contribute to the appearance of the wax. A representative directional source can be used for this scenario in the downward direction. However, this will incorrectly illuminate the wax melted at the bottom with high inaccuracies. Furthermore, the other scenes demonstrate errors due to the model being unable to account for the indirect illumination from the floor beneath.

Our model consistently outperforms the competition for a wide variety of object appearances substantiated quantitatively by the ꟻLIP mean on the bottom left of the images and the ꟻLIP error map on the bottom right of these images. Figure 8 shows an ablation study for the N in our formulation. Each N corresponds to five additional dimensions for the network input (two for $\vec{\omega}_i$ and three for $\boldsymbol{x}_i$). The study suggests that the model performs best when $N = 32$ or $N = 64$. As we use a 3-layers MLP for all our experiments, the higher error rate with $N \geq 128$ suggests that the function becomes too complicated to be learned with this network. This can be easily mitigated by using an MLP with more hidden layers at the cost of a more storage intensive method. Table 1 shows the training and inference timings of our network for the results in Figures 7 and

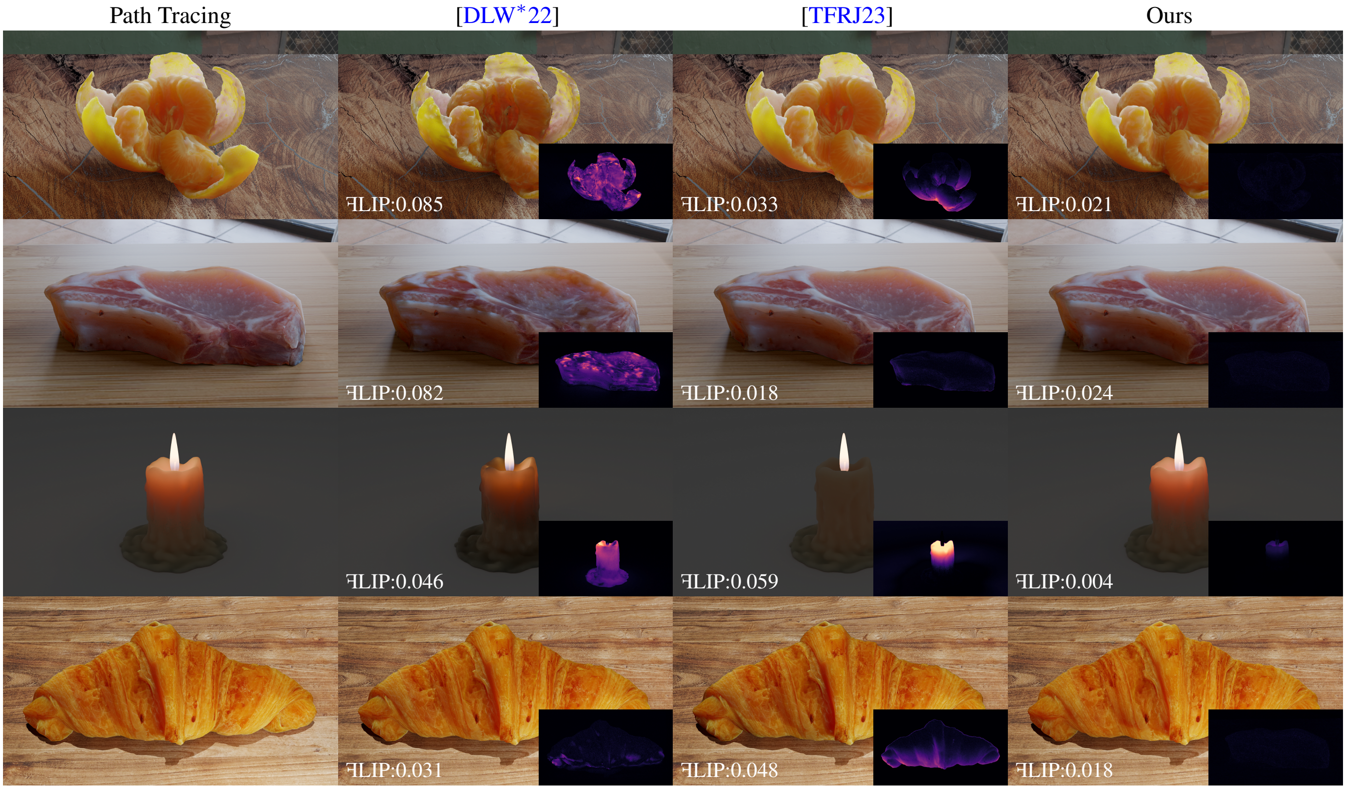

Figure 7: *Comparison with reference path tracing and recent work on representing the appearance of a heterogeneous translucent object: Deng et al. [DLW*22] using textured parameters for the standard dipole model, which was derived for homogeneous half-space materials, and TG et al. [TFRJ23] using a neural representation limited to distant lighting. Our model most closely matches the path traced references (see the inserted* ꟻLIP *mean and error maps). The* ꟻLIP *error maps clearly highlight the limitations of the models we compare with.*

9. Note that the *planar half-space* corresponds to every material in the 6th column of Figure 9.

5.2. Comparison with Analytic BSSRDF

To showcase our model for the planar half-space's ability to learn the BSSRDF we show the reflectance curve against several analytic models. For our reference volume path tracing we simulate using 8 dimensions, in particular, we collect radiance towards a given observation direction, $\vec{\omega}_o$. This will allow us a direct comparison of our network. The analytic models presented here do not depend on the observation direction and assume a diffuse reflectance as they are derived under the assumption of highly scattering media. This is one of the disadvantages of analytic models, which cannot capture effects due to changes in the direction of observation.

This is also seen in Figure 10, where our model captures the reflectance curves that analytic models cannot match as accurately. As a consequence, analytic models often lose some finer details and tend to blur out high-frequency details. Figure 9 compares renderings for a variety of materials with different albedos. Here, our model provides better results than the analytical models for the planar half-space when comparing the ꟻLIP scores. Rendering with a resolution of 1280×1280 took an average of 2-3 minutes with the analytical models to ensure a noise-free result, while photon beam diffusion [HCJ13] took around 10 minutes due to the need to sample the integral in the model, where we used 5 samples. On the other hand, our network rendered the images in less than two minutes and the path traced references took 1-3 hours.

5.3. Comparison with Denoised Volume Path Tracing

In Figure 11, we compare the rendering quality of our method with that of path tracing and denoising at equal rendering times. We use Intel Open Image Denoise [Áfr24] as the denoiser of our choice. Columns 2 and 4 show the output from the path tracer whereas columns 3 and 5 show their respective denoised results. As seen from the ꟻLIP errors on the images, our method outperforms path tracing in both scenarios. In path tracing (denoised), the denoiser tends to overcorrect the output to a higher radiance than expected due to the presence of firefly-like RGB noise, resulting in a worse ꟻLIP score compared to basic path tracing. In contrast, the absence of RGB noise in our method provides a significant advantage, allowing the denoiser to perform more accurate corrections.

5.4. Importance Sampling

Figure 12 demonstrates the effectiveness of our importance sampling module. This module needs to be retrained for every lighting

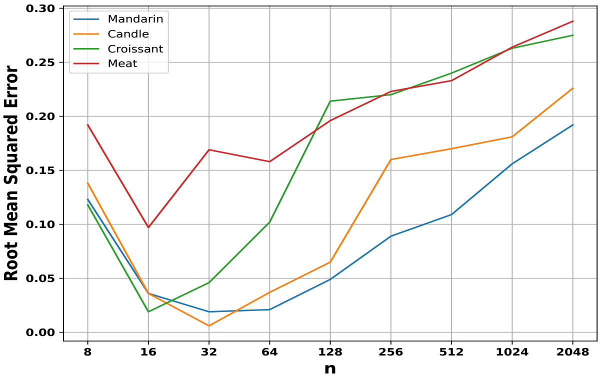

Figure 8: *We perform an accuracy analysis of our neural representation on all our models in Figure 7. This plot compares the model's accuracy in terms of root-mean-squared error (RMSE) with respect to N. The plot illustrates that the model fits better when $N = 32$ or $N = 64$. We use a 3-layers MLP for this experiments. With large N, the data complexity increase significantly which cannot be approximated with a 3-layers neural network.*

environment. However, due to the small size of the network and the small dataset obtained without path tracing, the network converges to acceptable accuracy in 100 iterations which takes less than 5 seconds on a GPU. We easily save the cost of this training through convergence with fewer samples per pixel. The sampled points, however, may not reside exactly on the surface. Our mitigation strategy was to replace each sample with a uniformly sampled point in the triangle with the closest barycenter. This is computationally expensive and might not be necessary. Figure 3 shows our sampled points with respect to the geometry and includes the root-mean-squared deviation (RMSD) of the sampled positions from their nearest barycenters. This RMSD is an upper bound of the error and because the object surfaces have many triangles, the RMSD demonstrates that the sampled points are very close to the surface. We have tested renderings without projection to the surface, and as long as a point is not a significant outlier, the appearance representation module can still work accurately with the points that are not exactly on the surface.

6. Discussion

The main limitation of our work is that each appearance representation network is trained and tied to its set of optical properties (and geometry for the *full geometry* formulation). A small adjustment of either requires training a new network or updating the network with new data. Even with the benefit of faster rendering times compared to analytic models, training a network for each set of optical properties can be time-consuming. One possibility is to incorporate a latent space for either geometry or optical properties into the network, which can be sampled. Such an approach was explored by Vicini et al. [VKJ19] for homogeneous materials and local geometry. Exploring similar approaches for heterogeneous and full geometry is interesting future work.

Although our importance sampling is a definite one-to-one mapping from a 3D Gaussian to the real distribution, this does not ensure that points are on the object's surface. Our projection method is simply to find the closest triangle and sample a point in it. While we have not found any issue with this, it is computationally expensive and could be replaced with a better projection technique (using for instance the gradient of a signed distance field). Even better, the projection is not necessary if the appearance representation network is robust enough. We use projection to make sure that this is not a source of error in the appearance representation, but we suggest a more comprehensive analysis of whether the projection is even necessary.

7. Conclusion

We presented a neural subsurface scattering representation technique. Our method faithfully and compactly represents the full 8-dimensional BSSRDF of a heterogeneous translucent object. We demonstrated how use of many small collections of sampled unidirectional paths through a volume is enough to capture such a light transport function in a tiny MLP. Furthermore, we extended our method to a half-space medium that simulates material appearance identical to classical analytic BSSRDF models where our method consistently produces accurate results with better performance, but re-training is needed when the optical properties change. Furthermore, we presented an importance sampling technique for finding relevant points of incidences to go along with our method.

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

[Áfr24] ÁFRA A. T.: Intel® Open Image Denoise. https://www.openimagedenoise.org, 2024. 9, 12

[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. 8

[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. 3

[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), pp. 12684–12694. doi:10.1109/ICCV48922.2021.01245. 3

[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. 3, 6

[Cha50] CHANDRASEKHAR S.: *Radiative Transfer*. Oxford University Press, 1950. 4

[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. 3

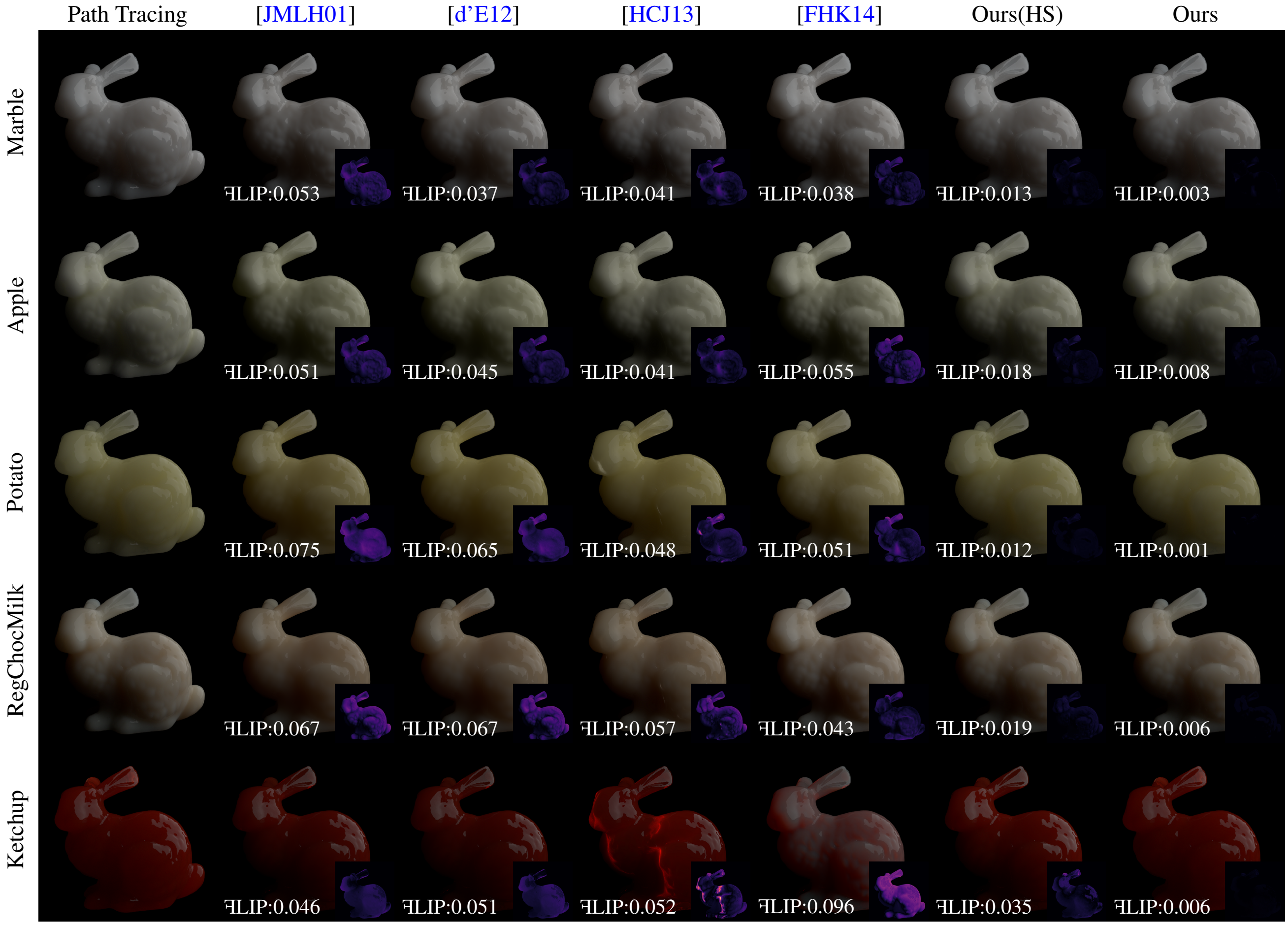

Figure 9: *Comparing the half-space (HS) and the full geometry versions of our method to various diffusion based approximations. Path traced single scattering was added to the models not including single scattering [JMLH01, d'E12]. At the bottom of the images, we include per pixel and mean ꟻLIP scores (lower is better) with respect to the reference path traced images. The half-space (HS) version of our model produces quality consistently better than the diffusion-based models, while the full geometry version matches the reference very well. The optical properties for the materials are marble, apple, potato, and ketchup [JMLH01] as well as regular chocolate milk [NGD*06].*

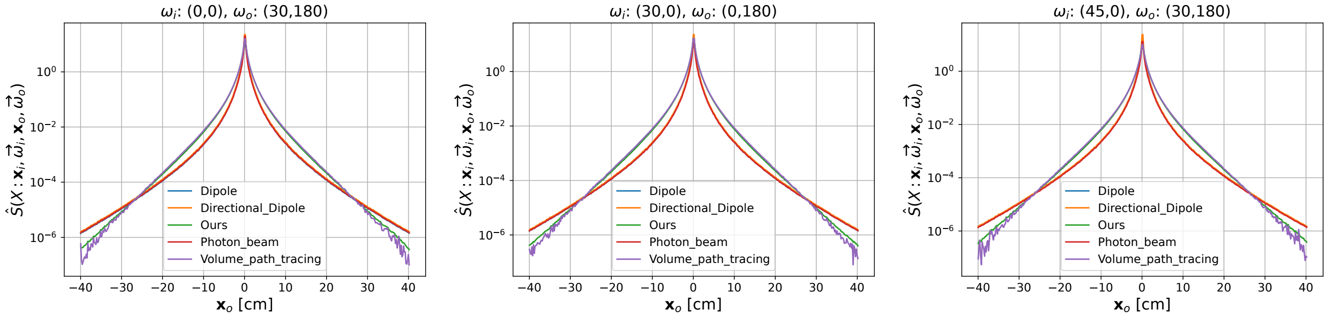

Figure 10: *BSSRDF reflectance curve comparison of our planar half-space formulation and analytic diffusion-based methods. This experiment was performed on a smooth plane without any surface reflections. We use isotropic scattering, extinction coefficient $\sigma_t = 2.41\,\text{cm}^{-1}$, a scattering albedo of 0.75, and a refractive index of $n = 1.3$. We compare our model with the standard dipole [JMLH01], the directional dipole [FHK14], photon beam diffusion [HCJ13] and standard volume path tracing. We selected the reflectance curves in a direction on the surface perpendicular to the plane of incidence. This is the reason for the symmetry of the curves on each side of the point of incidence. In this direction perpendicular to the plane of incidence, all the diffusion-based models exhibit similar issues, and we see a similar trend for more oblique angles of incidence.*

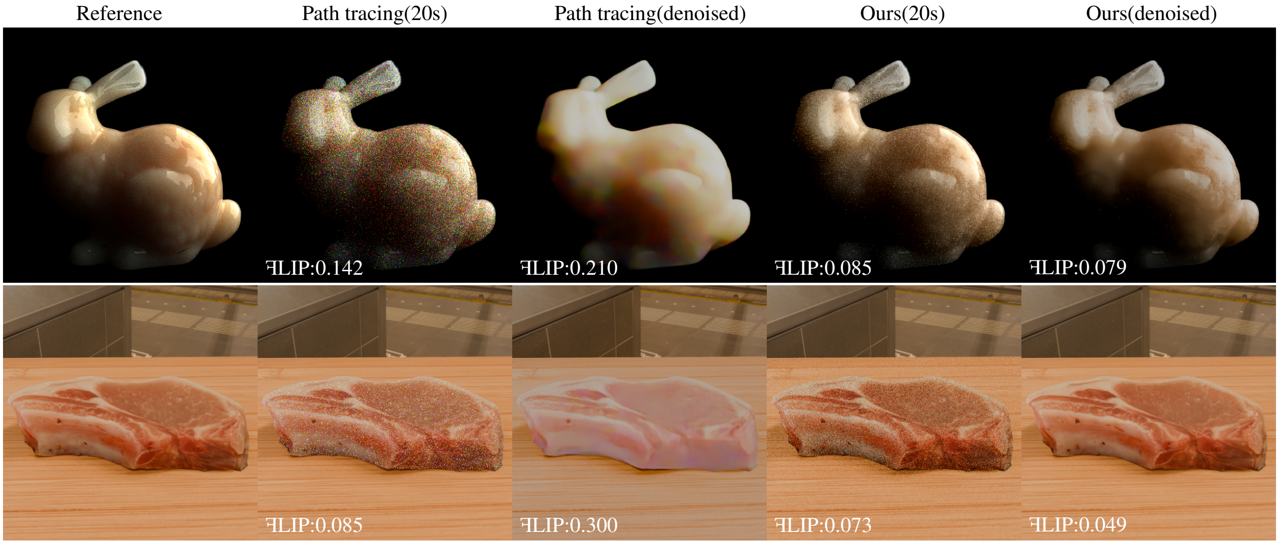

Figure 11: *Comparing rendering quality at equal render times (20 seconds). We use Intel Open Image Denoise [Áfr24] for denoising. Our representation does not produce any color noise which is advantageous for denoisers as compared with volume path tracing. The denoiser overestimates the color output due to the RGB noise which in turn produces perceptually worse results than volume path tracing as affirmed by the ꟻLIP scores at the bottom left of the images.*

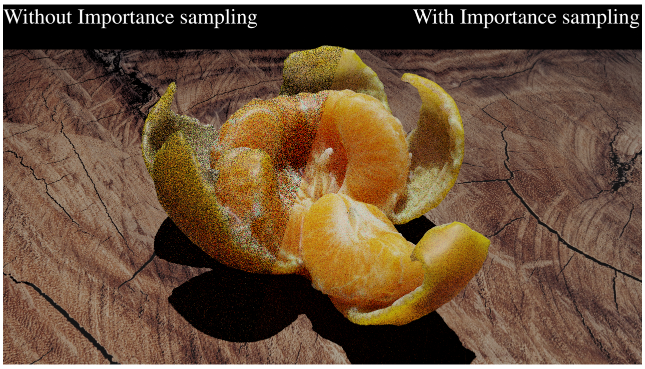

Figure 12: *Comparing the effectiveness of our representation network for a 1 $(\boldsymbol{x}_o, \vec{\omega}_o)$-sample per pixel rendering of the mandarin without our importance sampling (left) and with our importance sampling (right). The scene is lit by a single directional light and our importance sampling module provides sampling positions that have visibility to the light source, thereby providing a better estimate of the appearance.*

[d'E12] d'Eon E.: A better dipole. Technical report, 2012. URL: https://eugenedeon.com/pdfs/betterdipole.pdf. 11

[dI11] d'Eon E., Irving G.: A quantized-diffusion model for rendering translucent materials. *ACM Transactions on Graphics 30*, 4 (July 2011), 56:1–56:14. doi:10.1145/2010324.1964951. 3

[DJ05] Donner C., Jensen H. W.: Light diffusion in multi-layered translucent materials. *ACM Transactions on Graphics 24*, 3 (2005), 1032–1039. doi:10.1145/1073204.1073308. 3

[DJ07] Donner C., Jensen H. W.: Rendering translucent materials using photon diffusion. In *Rendering Techniques (EGSR)* (2007), pp. 243–251. doi:10.2312/EGWR/EGSR07/243-251. 3

[DKB15] Dinh L., Krueger D., Bengio Y.: Nice: Non-linear independent components estimation. In *International Conference on Learning Representations (ICLR)* (2015). doi:10.48550/arXiv.1410.8516. 5, 6

[DLR*09] Donner C., Lawrence J., Ramamoorthi R., Hachisuka T., Jensen H. W., Nayar S.: An empirical BSSRDF model. *ACM Transactions on Graphics 28*, 3 (2009), 30:1–30:10. doi:10.1145/1531326.1531336. 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. 3, 8, 9

[DVGNK99] Dana K. J., Van Ginneken B., Nayar S. K., Koenderink J. J.: Reflectance and texture of real-world surfaces. *ACM Transactions On Graphics 18*, 1 (1999), 1–34. doi:10.1145/300776.300778. 3

[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. 3

[FHK14] Frisvad J. R., Hachisuka T., Kjeldsen T. K.: Directional dipole model for subsurface scattering. *ACM Transactions on Graphics 34*, 1 (December 2014), 5:1–5:12. doi:10.1145/2682629. 3, 11

[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. 4

[FWH*23] Fan J., Wang B., Hasan 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. 3, 4

[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

[HCJ13] Habel R., Christensen P. H., Jarosz W.: Photon beam diffusion: a hybrid Monte Carlo method for subsurface scattering. In *Eurographics Symposium on Rendering (EGSR '13)* (2013), Eurographics Association, pp. 27—-37. doi:10.1111/cgf.12148. 3, 9, 11

[HMBVR05] Haber T., Mertens T., Bekaert P., Van Reeth F.: A computational approach to simulate subsurface light diffusion in arbitrarily shaped objects. In *Graphics Interface (GI)* (2005), pp. 79–86. doi:10.5555/1089508.1089522. 3

[HZE*19] Herholz S., Zhao Y., Elek O., Nowrouzezahrai D., Lensch H. P. A., Křivánek J.: Volume path guiding based on zero-variance random walk theory. *ACM Transactions on Graphics 38*, 3 (2019), 25:1–25:19. doi:10.1145/3230635. 2

[JC98] Jensen H. W., Christensen P. H.: Efficient simulation of light transport in scenes with participating media using photon maps. In *SIGGRAPH '98* (1998), ACM, p. 311–320. doi:10.1145/280814.280925. 2

[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. 3, 8, 11

[KGH*14] Křivánek J., Georgiev I., Hachisuka T., Vévoda P., Šik M., Nowrouzezahrai D., Jarosz W.: Unifying points, beams, and paths in volumetric light transport simulation. *ACM Transactions on Graphics 33*, 4 (2014), 103:1–103:13. doi:10.1145/2601097.2601219. 2

[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. 2

[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. 3

[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

[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. 3

[KWM*22] Kuznetsov A., Wang X., Mullia K., Luan F., Xu Z., Hasan M., Ramamoorthi R.: Rendering neural materials on curved surfaces. In *SIGGRAPH 2022 Conference Proceedings* (2022), ACM, pp. 9:1–9:9. doi:10.1145/3528233.3530721. 3

[LGB*02] Lensch H. P. A., Goesele M., Bekaert P., Kautz J., Magnor M. A., Lang J., Seidel H.-P.: Interactive rendering of translucent objects. In *Pacific Conference on Computer Graphics and Applications (PCCGA)* (2002), IEEE, pp. 214–224. doi:10.1109/PCCGA.2002.1167862. 3, 6

[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. 3

[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. 3

[LW96] Lafortune E. P., Willems Y. D.: Rendering participating media with bidirectional path tracing. In *Rendering Techniques '96 (EGWR)* (1996), Springer, pp. 91–100. doi:10.1007/978-3-7091-7484-5_10. 2

[MHD16] Meng J., Hanika J., Dachsbacher C.: Improving the Dwivedi sampling scheme. *Computer Graphics Forum 35*, 4 (2016), 37–44. doi:10.1111/cgf.12947. 2

[MMR*19] Müller T., McWilliams B., Rousselle F., Gross M., Novák J.: Neural importance sampling. *ACM Transactions on Graphics 38*, 5 (2019), 145:1–145:19. doi:10.1145/3341156. 5

[MST*20] Mildenhall B., Srinivasan P. P., Tancik M., Barron J. T., Ramamoorthi R., Ng R.: NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)* (2020), Springer, pp. 405–421. doi:10.1007/978-3-030-58452-8_24. 2, 3

[NGD*06] Narasimhan S. G., Gupta M., Donner C., Ramamoorthi R., Nayar S. K., Jensen H. W.: Acquiring scattering properties of participating media by dilution. *ACM Transactions on Graphics 25*, 3 (2006), 1003–1012. doi:10.1145/1141911.1141986. 11

[NGHJ18] Novák J., Georgiev I., Hanika J., Jarosz W.: Monte Carlo methods for volumetric light transport simulation. *Computer Graphics Forum 37*, 2 (2018), 551–576. doi:10.1111/cgf.13383. 2, 4

[Nic63] Nicodemus F. E.: Radiance. *American Journal of Physics 31*, 5 (1963), 368–377. doi:10.1119/1.1969512. 4

[NSJ14] Novák J., Selle A., Jarosz W.: Residual ratio tracking for estimating attenuation in participating media. *ACM Transactions on Graphics 33*, 6 (2014), 179:1–179:11. doi:10.1145/2661229.2661292. 2

[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 (July 2010), 66:1–66:13. doi:10.1145/1778765.1778803. 7

[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. 7

[PKK00] Pauly M., Kollig T., Keller A.: Metropolis light transport for participating media. In *Rendering Techniques 2000 (EGWR)* (2000), Springer, pp. 11–22. doi:10.1007/978-3-7091-6303-0_2. 2

[PN19] Panin M., Nikolenko S.: Faster RPNN: Rendering clouds with latent space light probes. In *SIGGRAPH Asia 2019 Technical Briefs* (2019), ACM, p. 21–24. doi:10.1145/3355088.3365150. 2

[Pre57] Preisendorfer R. W.: A mathematical foundation for radiative transfer theory. *Journal of Mathematics and Mechanics 6*, 6 (1957), 685–730. 4

[Pre65] Preisendorfer R. W.: *Radiative Transfer on Discrete Spaces*. Pergamon Press, 1965. doi:10.1016/C2013-0-05368-6. 4, 5

[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 (July 2006), 746–753. doi:10.1145/1141911.1141950. 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. 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. 2

[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. 3

[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. 3

[SSWN14] Sheng Y., Shi Y., Wang L., Narasimhan S. G.: Translucent radiosity: efficiently combining diffuse inter-reflection and subsurface scattering. *IEEE Transactions on Visualization and Computer Graphics 20*, 7 (2014), 1009–1021. doi:10.1109/TVCG.2013.256. 3, 6

[Sta95] Stam J.: Multiple scattering as a diffusion process. In *Rendering Techniques '95 (EGWR)* (1995), Springer, pp. 41–50. doi:10.1007/978-3-7091-9430-0_5. 3

[TFRJ23] TG T., Frisvad J. R., Ramamoorthi R., Jensen H. W.: Neural BSSRDF: Object appearance representation including heterogeneous subsurface scattering. arXiv:2312.15711 [cs.GR], 2023. doi:10.48550/arXiv.2312.15711. 2, 3, 4, 8, 9

[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. *Advances in Neural Information Processing Systems (NeurIPS* (2020), 7537–7547. doi:10.48550/arXiv.2006.10739. 2

[TWL*05] Tong X., Wang J., Lin S., Guo B., Shum H.-Y.: Modeling and rendering of quasi-homogeneous materials. *ACM Transactions on Graphics 24*, 3 (2005), 1054–1061. doi:10.1145/1073204.1073311. 3

[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. 3, 10

[WGH19] Wang B., Ge L., Holzschuch N.: Precomputed multiple scattering for rapid light simulation in participating media. *IEEE Transactions on Visualization and Computer Graphics 26*, 7 (2019), 2456–2470. doi:10.1109/TVCG.2018.2890466. 3

[WKB12] Walter B., Khungurn P., Bala K.: Bidirectional lightcuts. *ACM Transactions on Graphics 31*, 4 (2012), 59:1–59:11. doi:10.1145/2185520.2185555. 2

[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. 3

[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. 3

[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. 3

[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. 5

[YGF*23] Yu H.-X., Guo M., Fathi A., Chang Y.-Y., Chan E. R., Gao R., Funkhouser T., Wu J.: Learning object-centric neural scattering functions for free-viewpoint relighting and scene composition. *Transactions on Machine Learning Research*, 5 (2023). doi:10.48550/arXiv.2303.06138. 3

[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. 3

[YZXW12] Yan L.-Q., Zhou Y., Xu K., Wang R.: Accurate translucent material rendering under spherical gaussian lights. *Computer Graphics Forum 31*, 7 (2012), 2267–2276. doi:10.1111/j.1467-8659.2012.03220.x. 3

[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. 3, 4

[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. 3

[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 (December 2021), 237:1–237:18. doi:10.1145/3478513.3480496. 3

[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. 3