Towards Learning-based Inverse Subsurface Scattering

Chengqian Che, Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas

Abstract—Given images of translucent objects, of unknown shape and lighting, we aim to use learning to infer the optical parameters controlling subsurface scattering of light inside the objects. We introduce a new architecture, the inverse transport network (ITN), that aims to improve generalization of an encoder network to unseen scenes, by connecting it with a physically-accurate, differentiable Monte Carlo renderer capable of estimating image derivatives with respect to scattering material parameters. During training, this combination forces the encoder network to predict parameters that not only match groundtruth values, but also reproduce input images. During testing, the encoder network is used alone, without the renderer, to predict material parameters from a single input image. Drawing insights from the physics of radiative transfer, we additionally use material parameterizations that help reduce estimation errors due to ambiguities in the scattering parameter space. Finally, we augment the training loss with pixelwise weight maps that emphasize the parts of the image most informative about the underlying scattering parameters. We demonstrate that this combination allows neural networks to generalize to scenes with completely unseen geometries and illuminations better than traditional networks, with 38.06% reduced parameter error on average.

Index Terms—subsurface scattering, inverse scattering, differentiable rendering, inverse transport networks

1 Introduction

Translucent materials are everywhere around us, ranging from biological tissues to many industrial chemicals, and from the atmosphere and clouds to minerals. The common cause of the characteristic appearance of all these classes of materials is *subsurface scattering*: As photons reach the surface of a translucent object, they continue traveling in its interior, where they scatter, potentially multiple times, before reemerging outside the object.

The ubiquity of translucency has motivated decades of research across numerous scientific fields on problems relating to subsurface scattering. Broadly speaking, we can break these problems down into two categories. The first category is forward scattering problems, which attempt to predict the appearance of a translucent object, assuming that the optical parameters controlling scattering of light at its interior are known. Computer vision and computer graphics offer an array of algorithms for solving this problem, known in this literature as *volume rendering*, including Monte Carlo rendering algorithms that can reproduce translucent appearance in a physically-accurate way [1].

The second category, and the focus of this paper, is inverse scattering problems: Given images of a translucent object, they attempt to predict its underlying scattering parameters. Inverse scattering is an active research topic in many sciences outside of computer vision and computer graphics, including medical imaging, remote sensing, and material science. The fundamental challenge in inverse scattering is the extremely multi-path and multi-bounce nature of light propagation inside scattering volumes. The complexity of volume light transport makes inverse scattering a difficult problem even in the case where an object is characterized by a single set of, spatially-constant, material parameters (*homogeneous* scattering).

Among existing approaches for inverse scattering, many are based on simplifying assumptions about volume light transport, such as single scattering (all photons scatter once) and diffusion (all photons scatter a very large number of times). These assumptions limit the applicability of these methods to very optically-thin and thick materials [2], [3], excluding large classes of important *turbid* materials. Alternatively, recent years have seen the development of general-purpose inverse scattering techniques, which combine *analysis by synthesis* and Monte Carlo volume rendering in order to accurately estimate material parameters without the need for simplifications [4], [5], [6], [7], [8]. Despite their broad applicability, these techniques can be prohibitively computationally expensive: processing measurements of a new material often requires performing hundreds of expensive Monte Carlo rendering operations.

We investigate the use of deep learning techniques for inverse scattering problems, as a means to address the computational challenges of analysis by synthesis, while maintaining its broad applicability. We are inspired by recent successes of such techniques in other *inverse rendering* problems [9], [10], such as inferring shape, reflectance, and illumination from images [11], [12], [13], [14]. Despite these successes, the use of neural networks for inverse scattering remains unexplored, and we take first steps in this direction.

We begin by proposing a physics-aware learning pipeline that we term *inverse transport networks* (ITN), which aims to combine the computational efficiency of learning-based approaches with the generality of analysis by synthesis approaches for inverse scattering. Taking inspiration from recent work on combining physics and learning [11], [12], [13], [14], [15], these neural networks are trained to produce output parameters that not only match groundtruth values, but also reproduce the input images when used as input to a forward physics-based renderer. To be

- *C. Che and I. Gkioulekas are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 15213. E-mail: cche@andrew.cmu.edu*
- *F. Luan and K. Bala are with the Computer Science Department, Cornell University, Ithaca, NY 14850.*
- *S. Zhao is with the Computer Science Department, University of California Irvine, Irvine, CA 92697.*

able to train these neural networks efficiently, we pair them with a new *efficient* and *physically-accurate* Monte Carlo differentiable rendering engine [5], [6], [7], [16]. We further tailor these neural networks towards inverse scattering, by taking into account results from the radiative transfer literature, characterizing the conditions under which different scattering materials can produce similar translucent appearance [17], [18]. We introduce ways for making our networks robust to these ambiguities, including the use of non-linear material parameterizations, and weight maps emphasizing pixels where these ambiguities are weaker. We demonstrate the effectiveness of our networks in experiments on synthetic and real datasets, where we show that our networks can use a single uncalibrated (completely unknown shape and illumination) image input, to produce material parameter estimates that are on average 38.06% more accurate than those produced by baseline regression networks. Furthermore, images rendered with our predictions are on average 53.82% closer to the groundtruth. We release all of our implementations and datasets, to facilitate reproducibility and follow-up research [19].

2 Related work

Subsurface scattering. Forward scattering, also known as volume rendering, algorithms in computer vision and computer graphics are predominantly based on the radiative transfer framework [20]. Existing techniques include Monte Carlo volume rendering and photon mapping algorithms [1]. Inverse scattering techniques include approaches relying on single-scattering approximations [2], [21], [22], [23], which are appropriate for optically thin media; as well as diffusion-based approaches [3], [24], [25], [26], [27], [28], [29], [30], suitable for optically thick media. Intermediate cases, so-called turbid scattering, can be tackled using techniques based on combinations of analysis by synthesis and differentiable Monte Carlo rendering, as discussed below. Recently, deep learning techniques have been used to accelerate forward scattering simulation [31], [32]. To the best of our knowledge, we are the first to consider deep learning techniques for the inverse scattering problem.

Analysis by synthesis in physics-based vision. Analysis by synthesis is a core methodology for recovering physical scene parameters from images, which conceptually comprises three steps: (i) formulate an approximate image formation (or forward rendering) model as a function of the scene parameters; (ii) analytically derive an expression for the derivative of the forward model with respect to those parameters; (iii) use gradient-based optimization to solve an analysis-by-synthesis objective comparing measured and synthesized images. This approach has been used to recover shape [33], [34], material [35], [36], [37], and illumination [38], independently or jointly [39], [40], [41], [42].

Differentiable rendering. Analysis by synthesis requires formulating a new forward model, as well as analytically computing its derivatives, specifically for each reconstruction problem. Differentiable renderers such as OpenDR [43] have been proposed to remove this obstacle, by providing a general-purpose framework that can be differentiated with respect to arbitrary scene parameters. To ensure analytical differentiability, these approaches use *approximate* forward models, ignoring complex light transport effects such as inter-reflections and subsurface scattering. This makes these methods inapplicable to situations where these effects are dominant. Differentiable Monte Carlo rendering algorithms overcome this limitation, by estimating derivatives of images while accounting for all light transport effects. These algorithms were first introduced in the context of differentiation with respect to scattering parameters, and used for accurate inverse scattering [4], [5], [6], [7], [8]. Since then, they have been extended to allow differentiation with respect to, and recovery of, arbitrary scene parameters, including surface reflectance [44], [45], geometry [46], and visibility and pose [16], [47], [48].

Combining deep learning with rendering. Recently, a number of works have proposed using renderers not for analysis-by-synthesis, but as parts of learning architectures. The most popular approach is to replace the decoder network in an auto-encoder pipeline [49], [50] with a rendering layer that takes as input the parameters predicted by the encoder and produces as output synthesized images. This encoder-renderer architecture was first proposed by Wu et al. [51], who used a non-photorealistic renderer to achieve categorical interpretability. The same conceptual architecture was later used, together with approximate (direct lighting) physics-renderers for inference of physical scene parameters such as surface normals, illumination, and reflectance [11], [12], [13], [14], [15], [52], [53], [54], [55], [56], [57], [58]. Inspired from these works, we apply the encoder-renderer architecture to the problem of inverse scattering, using for the first time a physically-accurate Monte Carlo differentiable renderer instead of an approximate one. Finally, differentiable Monte Carlo rendering has also been combined with neural networks in the context of discovering adversarial example scenes for classification tasks [16].

Compared to this prior work, we show a new use of differentiable renderers, as regularization during the training of neural networks for inverse scattering tasks. Despite their low dimensionality, these tasks remain challenging due to the complexity of subsurface light transport. Inverse scattering is of high relevance to several other sciences (medicine, remote sensing, material science). By combining neural networks with differentiable rendering, we take first steps towards developing robust, physics-aware, learning-based approaches for this problem.

3 Background on Inverse Scattering

Problem setting. We are interested in the problem of *homogeneous inverse scattering*: Given an image of a translucent object, of potentially unknown shape and lighting, we aim to determine the optical material parameters that control the scattering of light inside this object. These parameters are:

- The *extinction coefficient* σ_t is the scalar optical density of the material, controlling the average distance between consecutive volume events.
- The *volumetric albedo* α is the scalar probability of whether photons are scattered or absorbed at volume events.
- The *phase function* f_p is the spherical probability distribution controlling the direction scattered photons continue to travel towards.

The phase function is typically assumed to be only a function of the inner product between incoming and outgoing directions. The first moment of the phase function, or *average cosine*, $\bar{c} = 2\pi \int_{-1}^{1} c f_p(c)\,\mathrm{d}c$, $-1 \leq \bar{c} \leq 1$, is commonly used to characterize a material as predominantly forward-scattering ($\bar{c} > 0$), backward-scattering ($\bar{c} < 0$), or isotropic ($\bar{c} = 0$). From the above parameters, we can also derive the *scattering coefficient* $\sigma_s = \alpha \cdot \sigma_t$ and *absorption coefficient* $\sigma_a = (1 - \alpha) \cdot \sigma_t$, which

describe how much light is scattered and absorbed, respectively, at each scattering event. In general, all these parameters can be spatially varying, but in our setting we assume they are constant everywhere inside the object (homogeneous scattering). Throughout the paper, we will be using different subsets of the above parameters, as well as certain non-linear functionals, to characterize scattering. We will be denoting each material as $\boldsymbol{\pi}$, with the corresponding parameterization inferred from context. We discuss specific parameterizations in Section 5.

The primary difficulty of the inverse scattering problem lies in the complexity of the underlying *volumetric light transport* physics: Each photon propagating inside a scattering medium undergoes a random walk, controlled non-linearly by the medium's parameters. These random walks, described by the radiative transfer equation [20] typically involve more than one bounce. In turn, a radiometric detector capturing an image of such an object accumulates a large number of photons, each performing a different random walk. As a consequence of this extremely multi-path and multi-bounce light transport, images of translucent objects are highly non-linear functions of the underlying material parameters. We will represent this complex image formation process using the operator $\mathcal{T}(\boldsymbol{\pi})$, where $\boldsymbol{\pi}$ are the material parameters. We note that $\mathcal{T}$ is also a function of other scene parameters, such as shape and illumination; we omit this dependence for notational simplicity, and to focus on the material parameters we are interested in recovering.

In certain cases, we can simplify this image formation model by assuming that each photon only bounces either once or a very large number of times inside the object. These approximations, known as *single scattering* and *diffusion* respectively, are of limited applicability, as they are only accurate for very optically thin [2] or thick [3] materials. Additionally, the diffusion approximation cannot be used near thin geometric features such as sharp edges.

Analysis by synthesis. The shortcomings of these approximations have motivated the development of general-purpose inverse scattering techniques that accurately model the full complexity of volumetric light transport [4], [5], [6], [7], [8]. These techniques operate within the framework of *analysis by synthesis*, also known in computer graphics as *inverse rendering*. Given image measurements I, we search for parameters $\boldsymbol{\pi}$ that, when used to synthesize images, can closely match the measurements. This approach can be succinctly written as the following optimization problem:

$$\hat{\boldsymbol{\pi}} = \operatorname{argmin}_{\boldsymbol{\pi}} \left\| I - \mathcal{T}(\boldsymbol{\pi}) \right\|^2. \tag{1}$$

This procedure can be used for inverse scattering in objects of arbitrary *known* shape and lighting. This is thanks to the advent of graphics algorithms that can accurately simulate the full complexity of volumetric light transport. Besides traditional *forward rendering* algorithms that synthesize images as functions of material parameters $\boldsymbol{\pi}$ [1], recent years have seen the development of *differentiable rendering* algorithms that compute image derivatives with respect to these parameters, $\partial \mathcal{T}(\boldsymbol{\pi}) / \partial \boldsymbol{\pi}$ [4], [5], [6], [7], [16]. Differentiable rendering algorithms can greatly accelerate analysis by synthesis, by enabling the use of gradient descent algorithms for solving the optimization problem (1).

Despite these advances, performing inverse scattering by analysis by synthesis remains challenging in many situations. First, solving optimization (1), even with gradient descent, is computationally intensive, requiring performing hundreds or thousands of expensive rendering operations. Second, the use of gradient descent means that the analysis by synthesis optimization is susceptible to local minima in the loss function of Equation (1). This issue is particularly pronounced in inverse scattering, where the highly-nonlinear function $\mathcal{T}(\boldsymbol{\pi})$ results in large classes of different material parameters $\boldsymbol{\pi}$ that can produce similar images I. These scattering parameter ambiguities are known as *similarity relations* [17], [18]. Third and last, performing inverse scattering requires accurate calibration of ancillary scene parameters such as shape, illumination, and camera pose, which is not possible except in controlled lab environments.

We aim to overcome these challenges by investigating the use of data-driven algorithms for the inverse scattering problem. In particular, in Section 4, we discuss how to alter the training procedure of neural networks, to produce networks that, at test time, can use a single uncalibrated input image to produce material estimates $\boldsymbol{\pi}$ that are close to those we would obtain from analysis by synthesis. Then, in Section 5, we discuss design choices, inspired from the physics of scattering, that help these networks overcome ambiguities due to similarity relations.

4 Inverse Transport Networks

Supervised learning provides an alternative to the analysis by synthesis methodology for inverse rendering problems, and has previously been successful for tasks such as reflectance, illumination, and shape inference [11], [12], [13], [14]. These prior successes motivate us to investigate the use of learning techniques for the inverse scattering problem.

Supervised learning assumes availability of a training set of image measurements $\{I_d\}_{d=1}^D$ and corresponding groundtruth material parameters $\{\boldsymbol{\pi}_d\}_{d=1}^D$. Given a training dataset, learning techniques use empirical risk minimization to train a parametric regression model $\mathcal{N}[\boldsymbol{w}]$, e.g., a neural network, that directly maps images to parameters:

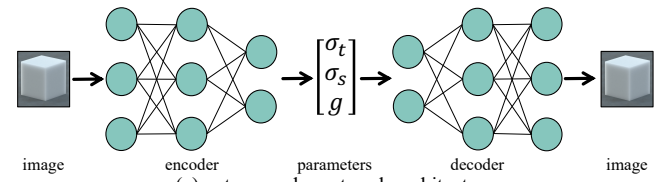
$$\hat{\boldsymbol{w}} = \operatorname{argmin}_{\boldsymbol{w}} \sum_{d=1}^{D} \left\| \boldsymbol{\pi}_d - \mathcal{N}[\boldsymbol{w}](I_d) \right\|^2. \tag{2}$$

The trained network $\mathcal{N}[\hat{\boldsymbol{w}}]$ can be used to efficiently obtain parameter estimates $\hat{\boldsymbol{\pi}}$ for new images I, through *forward pass* operations: $\hat{\boldsymbol{\pi}} = \mathcal{N}[\hat{\boldsymbol{w}}](I)$. This is in contrast with analysis by synthesis, which requires solving the expensive optimization problem (1) for every new input image. Additionally, the trained network can be used with images where other scene parameters are completely uncalibrated.

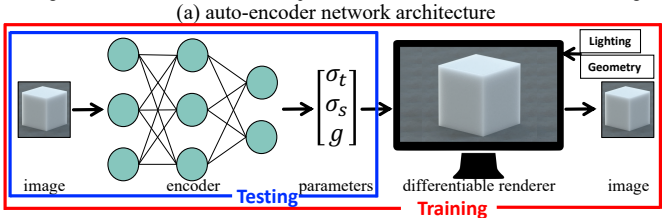
These advantages of supervised techniques come with the caveat that it is difficult to guarantee the accuracy of the estimates $\hat{\boldsymbol{\pi}}$ obtained for images of scenes that are not well represented in the training set. Given the highly nonlinear mapping $\mathcal{T}$ from scene to images in the case of subsurface scattering, it is challenging to train networks that generalize well to scenes of, e.g., very different shape or illumination

In order to combine the complementary advantages of learning and analysis by synthesis, we propose to regularize the training loss function (2) with a term that closely resembles the loss function (1) of analysis by synthesis:

$$\hat{\boldsymbol{w}} = \operatorname{argmin}_{\boldsymbol{w}} \sum_{d=1}^{D} \Big[\underbrace{\left\| \boldsymbol{\pi}_d - \mathcal{N}[\boldsymbol{w}](I_d) \right\|^2}_{\text{supervised loss}} + \lambda \underbrace{\left\| I_d - \mathcal{T}(\mathcal{N}[\boldsymbol{w}](I_d)) \right\|^2}_{\text{regularization}} \Big]. \tag{3}$$



(a) auto-encoder network architecture



(b) inverse transport network architecture

Fig. 1: **Inverse transport networks:** (a) Traditional autoencoders use two networks, encoder and decoder, to learn to predict parameters from images. (b) Inverse transport networks replace the decoder with a differentiable Monte Carlo renderer, to improve the generalization and physical accuracy of the predictions. During training, the renderer is provided with the material parameter output by the encoder network, as well as with groundtruth geometry and illumination, to perform forward and backward evaluations of an additional appearance-matching regularization term used to learn the network weights. During testing, the encoder network is used on its own, without the renderer: It takes as input a single, fully uncalibrated (unknown geometry and illumination) image, and produces as output a set of material parameters.

The regularization term in Equation (3) forces the neural network to predict parameters $\boldsymbol{\pi}_d$ that not only match the groundtruth, but also can be used with forward rendering to reproduce the input images. This has two desirable effects: First, the parameters predicted by the network are likely to be close to what would have been obtained from analysis by synthesis, as the regularization term in Equation (3) is equivalent to the analysis by synthesis loss (1). Second, the regularization term forces the neural network $\mathcal{N}[\hat{\boldsymbol{w}}]$ to be approximately equal to the inverse of the volumetric light transport operator $\mathcal{T}$, that is, $\mathcal{N}[\hat{\boldsymbol{w}}] \approx \mathcal{T}^{-1}$. Given that $\mathcal{T}$ models the physics of subsurface scattering for scenes of arbitrary geometries and illumination, we expect the resulting neural network to generalize well to novel scenes. We refer to networks trained using the loss (2) as *regressor networks* (RN), and to networks trained using (3) as *inverse transport networks* (ITN), based on their above-discussed property.

Relationship to prior work. Regularization similar to Equation (3) has previously appeared in two general forms. The first is autoencoder architectures [49], [50] that, in addition to the regressor (*encoder*) network $\mathcal{N}[\boldsymbol{w}]$ mapping images to parameters, use a second *decoder* network $\mathcal{D}[\boldsymbol{u}]$ that maps the parameters back to images. Then, the regularization term in Equation (3) is replaced with $\|I_d - \mathcal{D}[\boldsymbol{u}](\mathcal{N}[\boldsymbol{w}](I_d))\|^2$, and both the encoder and decoder networks are trained simultaneously, potentially without access to groundtruth parameters (self-supervised learning). These architectures are of great utility when inferring semantic parameters (e.g., a class label) of a scene, where there is generally no analytical model for the forward mapping of these parameters to images. However, when the unknowns $\boldsymbol{\pi}$ are scattering material parameters, autoencoder architectures do not take advantage of the rich knowledge we have about the physics governing the forward operator $\mathcal{T}$. Additionally, the forward mapping $\mathcal{D}[\boldsymbol{u}]$ may not generalize to novel scenes, as it is specific to the training dataset. Figure 1 compares the autoencoder and inverse transport architectures.

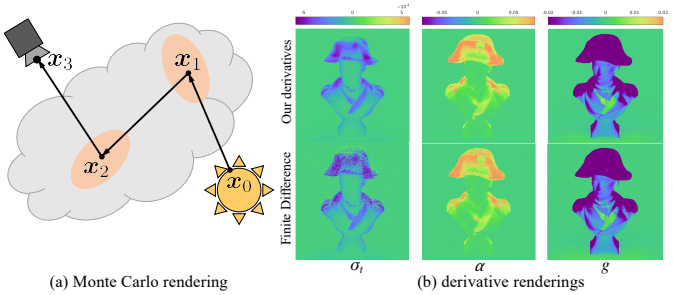
There are also networks that use regularization terms where the light transport operator $\mathcal{T}$ is replaced with an approximate rendering model [11], [12], [13], [14], [15]. These approximations generally use direct lighting models, where photons are assumed to only interact with the scene once between emission and detection (e.g., direct reflection without interreflections). Unfortunately, these networks have limited applicability to the case of inverse scattering, where the underlying physics are characterized by extremely multi-path, multi-bounce light transport. Inspired by these prior works, our ITNs are physics-aware learning pipelines that can be used even in the presence of these higher-order transport effects that are dominant in inverse scattering.

Training ITNs. The optimization problem (3) for ITN training is computationally challenging: Evaluating the operator $\mathcal{T}$ requires solving the radiative transfer equation [20]. In theory, training could be performed using algorithms such as REINFORCE [51], which do not require differentiating the regularization term and only employ graphics rendering algorithms for forward evaluations of $\mathcal{T}$. However, such algorithms are known to suffer from slow convergence.

Instead, we aim to optimize the loss (3) with state-of-the-art stochastic gradient descent algorithms [59]. This requires using *differentiable rendering* algorithms to estimate derivatives of $\mathcal{T}$ with respect to material parameters $\boldsymbol{\pi}$ in an unbiased manner. For this, we rely on prior work [4], [5], [6], [7] that devised Monte Carlo rendering algorithms for simulating these derivatives by simulating the full volumetric light transport in a *physically-accurate* way. These algorithms have subsequently been generalized to scene parameters such as reflectance [44], [45], geometry [46], and pose [16], [47], [48]. For completeness, we provide below an overview of the differentiable rendering formulation at the basis of our work. We note that, because we optimize over only material parameters, our differentiable rendering formulation is significantly simpler than that required for dealing with global geometry changes, and which has been developed extensively in recent works [16], [47], [48].

Figure 1 provides an overview of our pipeline at training and test time: During training, the network is connected to the differentiable renderer. The network takes as input a single, high-dynamic-range image, and produces as output a set of scattering material parameters. During training, the network is connected to the differentiable renderer. The renderer takes as input the parameters produced by the network, as groundtruth geometry and illumination, to compute values and gradients of the regularization term in Equation (3). As we discuss in Section 6, because we train the network using synthetic input images, the geometry and illumination are readily available. During testing, the network is used on its own, without the renderer. As our objective is to use the network on testing images that are completely uncalibrated, no geometry or illumination information is given as input to the network during either training or testing.

Differentiable Monte Carlo volume rendering. To keep the paper self-contained, we provide a brief overview of forward and differentiable rendering in the context of subsurface scattering. Our discussion largely follows [7]. The starting point for both types of rendering is the path integral formulation of light transport, which expresses the images captured by a radiometric



(a) Monte Carlo rendering (b) derivative renderings

Fig. 2: **Monte Carlo rendering:** (a) Monte Carlo forward rendering estimates radiometric measurements by randomly sampling light paths and aggregating their radiance contributions. By evaluating additional terms for each path, we can use the same procedure to estimate *derivatives* of measurements with respect to physical scene parameters. (b) Example renderings of the derivatives of a scene with subsurface scattering with respect to different material parameters. The top row shows derivatives estimated by our differentiable renderer, and the bottom row shows derivatives estimated using finite differences.

detector as integrals over the space of possible light paths [1]:

$$\mathcal{T}(\boldsymbol{\pi}) = \int_{\mathbb{P}} f[\boldsymbol{\pi}](\bar{\boldsymbol{x}})\, \mathrm{d}\bar{\boldsymbol{x}}. \tag{4}$$

The above integration is performed over the space $\mathbb{P}$ of all possible light paths of the form $\bar{\boldsymbol{x}} \equiv (\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_K)$, for any $K > 1$ and with $\boldsymbol{x}_k \in \mathbb{R}^3$ (for $k = 0, 1, \ldots, K$). For each such path, $\boldsymbol{x}_0$ is located on a light source, $\boldsymbol{x}_K$ on a sensor, and intermediate vertices $\boldsymbol{x}_k$ light-scene interactions via reflection, refraction, and subsurface scattering. The *throughput function* $f[\boldsymbol{\pi}]$ describes the amount of radiance contributed by a path as a function of the scene geometry, material, illumination, and detector.

By differentiating Equation (4) and rearranging throughput terms and their derivatives, we can obtain a similar path integral expression for the derivatives $\partial\mathcal{T}(\boldsymbol{\pi})/\partial\boldsymbol{\pi}$ of images with respect to the scattering parameters $\boldsymbol{\pi}$:

$$\partial\mathcal{T}(\boldsymbol{\pi})/\partial\boldsymbol{\pi} = \int_{\mathbb{P}} f[\boldsymbol{\pi}](\bar{\boldsymbol{x}})\, S[\boldsymbol{\pi}](\bar{\boldsymbol{x}})\, \mathrm{d}\bar{\boldsymbol{x}}. \tag{5}$$

Compared to Equation (4), the path integral for this case includes the *score function* $S[\boldsymbol{\pi}]$, that sums derivatives of the per-vertex throughput with respect to $\boldsymbol{\pi}$.

Monte Carlo rendering algorithms evaluate the integrals of Equations (4) and (5) using Monte Carlo integration: (i) paths $\{\bar{\boldsymbol{x}}_n : n = 1, \ldots, N\}$ are drawn from a probability density p over the path space $\mathbb{P}$; (ii) their throughputs $f_s[\boldsymbol{\pi}]$ are computed; and (iii) unbiased and consistent estimators of Equations (4) and (5) are formed as

$$\langle\mathcal{T}(\boldsymbol{\pi})\rangle = \frac{1}{N}\sum_{n=1}^{N} f[\boldsymbol{\pi}](\bar{\boldsymbol{x}}_n)/p(\bar{\boldsymbol{x}}_n), \tag{6}$$

$$\langle\partial\mathcal{T}(\boldsymbol{\pi})/\partial\boldsymbol{\pi}\rangle = \frac{1}{N}\sum_{n=1}^{N} \frac{f[\boldsymbol{\pi}](\bar{\boldsymbol{x}}_n) S[\boldsymbol{\pi}](\bar{\boldsymbol{x}}_n)}{p(\bar{\boldsymbol{x}}_n)}. \tag{7}$$

We use our own implementation of differentiable rendering: We integrated the Stan Math Library [60] for automatic differentiation of throughput terms, with the Mitsuba engine [61] for physically accurate Monte Carlo rendering. We use Mitsuba's volumetric path tracing algorithm to sample paths for forming the estimates of Equations (6) and (7). Even though our focus is on inverse scattering, our implementation is a general-purpose differentiable renderer that can compute derivatives for scene parameters such as normals, reflectance, and illumination. We verified correctness of our derivatives by comparing derivatives computed using finite differences. An example comparison is shown in Figure 2. Note that the finite-difference gradients required more than two million samples-per-pixel, compared to 16384 samples-per-pixel used by the differentiable renderer. This shows the critical performance advantages of using differentiable rendering instead of numerical differentiation, which have also been well-documented in the past [4], [5], [6], [7], [16], [47], [48]. Our differentiable renderer implementation is available on the project website [19].

Stochastic optimization. In addition to physical accuracy, Monte Carlo differentiable rendering provides computational advantages in the context of gradient-based optimization. In particular, training deep neural networks strongly relies on the ability to perform backpropagation in a *stochastic* manner, by computing derivatives of the loss function (2) using random subsets of the training set (*minibatches*). Changing the minibatch size allows controlling the tradeoff between the cost of gradient computations and the number of iterations for convergence [62], [63].

Monte Carlo differentiable rendering offers control over a similar capability: We can reduce the number of sampled paths to accelerate derivative computation, at the cost of increased variance. As the Monte Carlo derivative estimates are consistent and unbiased, we can use this to take advantage of the same convergence guarantees and tradeoffs as with stochastic backpropagation. Therefore, our Monte Carlo differentiable rendering engine is particularly well-suited for training of neural networks using state-of-the-art stochastic gradient descent algorithms [59].

Post-learning refinement. Our focus is on using inverse transport networks as an inference algorithm that can be used in place of analysis by synthesis optimization when the latter is not possible, e.g., when dealing with uncalibrated scenes of unknown geometry and illumination. We mention though, that inverse transport networks can be useful even when these scene parameters are calibrated and analysis by synthesis can be performed. In particular, the trained network $\mathcal{N}$ can be used to produce a first estimate of the unknown parameters $\boldsymbol{\pi}$ underlying an input image I. This estimate can be used to *warm-start* subsequent analysis by synthesis optimization, by serving as initialization for the gradient descent minimization of the analysis by synthesis loss (1) for the image I. The effect of this warm-starting procedure is that the analysis by synthesis optimization can converge much faster than if we had skipped the network-based estimation stage and used a random initial point. We expect the ITN architecture to be particularly effective for this kind of analysis by synthesis acceleration, given that the regularization term in its training loss function (3) encourages the network to produce estimates that are close to the analysis by synthesis solution. Additionally, the ITN-based initialization can help the analysis by synthesis minimization avoid local minima due to similarity relations.

5 Overcoming Similarity Relations

Similarity relations describe classes of material parameters that produce very similar appearance under certain geometry and lighting conditions [17], [18]. These relations are derived from the radiative transfer equation, and are well-studied in the subsurface scattering literature. We focus on *first-order* similarity relations, which are the most commonly-used class of material ambiguities: Two materials $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ are considered similar if they satisfy,

$$\sigma_a = \sigma'_a, \quad \sigma_s \cdot (1 - \bar{c}) = \sigma'_s \cdot (1 - \bar{c}'). \tag{8}$$

These ambiguities can be problematic for both analysis by synthesis and supervised learning techniques for inverse scattering. In the following, we discuss two strategies for ameliorating the negative effect of similarity relations.

Material parameterization. As discussed in Section 3, there are several redundant parameters that are typically used to characterize the space of scattering materials. Prior work has considered parameterizations in terms of non-linear functionals of these parameters that, when used with analysis by synthesis, reduce estimation errors due to similarity relations [18]. We take advantage of this prior work, and use these parameterizations in the loss functions (2) and (3). We parameterize the material space as $\boldsymbol{\pi} = \{\sigma_a, \sigma_s, \sigma_s^r\}$, where σ_s^r is the *reduced scattering coefficient*,

$$\sigma_s^r = \sigma_s \cdot (1 - \bar{c}). \tag{9}$$

Intuitively, the robustness of this parameterization is due to the fact that the reduced scattering coefficient exactly matches the second similarity relation equation (8). The same parameterizaion also arises when deriving the *reduced* scattering properties of diffusion-based subsurface scattering [3]. Throughout the rest of the paper, we refer to this as the *similarity-aware parameterization*. In Section 6, we compare this with other naive parameterizations in the context of supervised learning for inverse scattering.

Per-pixel weight maps. Similarity relations are derived under the assumption that photons perform a large number of scattering events inside an object. As a consequence, their accuracy is strongly-dependent on scene conditions such as illumination and shape. Specifically, at thin parts of the object or parts of the surface with sharp geometric features (e.g., geometric edges), two materials will have different appearance even if their scattering parameters satisfy the similarity relations. The importance of thin geometric features for translucent appearance is well-documented in the literature, even beyond similarity considerations. These features has been shown to provide rich information about the scattering parameters of an object [64], and to be important for the *perception* of translucency by humans [65], [66], [67].

Motivated by the above, we modify the regularization term in the training loss (3), to use a per-pixel weight map,

$$\textstyle\sum_{i,j} \left\| w_d^{ij} (I_d^{ij} - \mathcal{T}\left(\mathcal{N}\left[\boldsymbol{w}\right](I_d)\right)^{ij})^2 \right\|. \tag{10}$$

where the superscript ij indicates indexing an image at pixel coordinates $[i, j]$, and summation is done over all pixels. For each image I_d in the training dataset, the per-pixel weights are selected to emphasize pixels corresponding to parts of the object with thin geometry, where similarity relations are not accurate.

Determining optical thickness, that is, the average distance light travels inside the object, requires knowing the groundtruth shape and illumination. In lieu of these, we use a simple algorithm for generating a weight map from only the input image I_d: Considering that, in a textureless homogeneous material, all image-space edges correspond to geometric discontinuities, we first process the image I_d with an edge detector, then assign to each pixel $[i, j]$ a weight w_d^{ij} equal to its distance from the nearest edge. Figure 3 shows example weight maps created this way. As this weight map is computed from only the input image without requiring any additional information, we additionally provide it as an input to the network during both training and testing. Despite its simplicity, the figure and the results of Section 6 show that our algorithm is robust enough to produce meaningful weight maps resulting in significant performance improvements for a large variety of geometries.

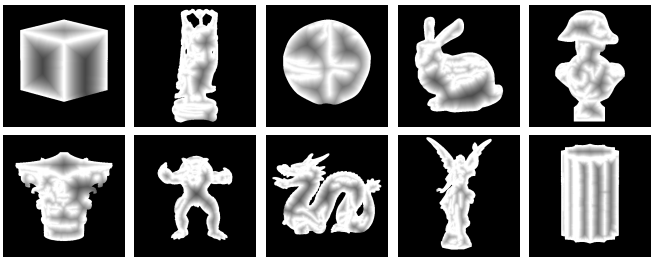


Fig. 3: **Weight maps:** We use per-pixel weights equal to each pixel's distance from the nearest image edge, in order to emphasize image pixels where similarity relations are violated.

6 Experiments

We evaluate the performance of different neural networks through experiments on simulated datasets and real images. We show additional results in the supplement.

Network details. We compare neural networks trained with five different loss functions. First, a regressor network (RN), trained using the purely supervised loss (2), and the *naive parameterization* $\boldsymbol{\pi} = \{\sigma_t, \alpha, \bar{c}\}$. Second, an RN using the similarity-aware parameterization of Section 5, $\boldsymbol{\pi} = \{\sigma_a, \sigma_s, \sigma_s^r\}$. Third, an inverse transport network (ITN), trained using the regularized loss (3), and the naive parameterization. Fourth, an ITN that uses the regularization (3) and the similarity-aware parameterization. Fifth and last, an ITN that uses the weighted regularization (10) and the similarity-aware parameterization.

All networks take as input a single high-dynamic-range image. For all networks, we use a state-of-the-art architecture for inverse rendering problems relating to homogeneous reflectance [11], [13]: Each network is composed of seven convolutional layers, and the size of the output channel for each layer is reduced to half the size of its input. The kernel size for the convolutional layer is 3 by 3, with a stride of 2 and padding of 1. Each convolutional layer is followed by a rectified linear unit (ReLU) and a max-pooling layer. A fully-connected layer is used at the end. We visualize this architecture in Figure 4. We select this architecture as it reflects the state-of-the-art in the supervised deep material task that is closest to ours: inferring homogeneous BRDF parameters (as far as we know, there is no prior work on estimating homogeneous subsurface scattering). The use of this architecture ensures that the RN with the naive parameterization can serve as meaningful baselines for evaluating the importance of our various innovations (similarity-aware parameterization, regularization using the differentiable Monte Carlo renderer, and weight map).

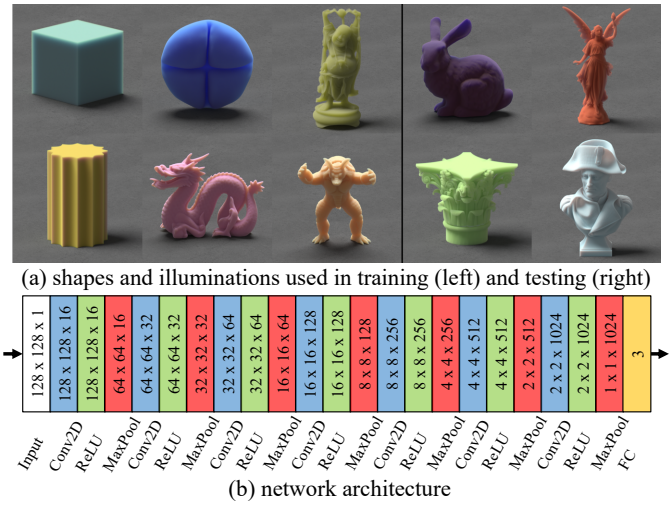
When training ITNs, we use as initialization an RN trained for a few epochs. We set λ in Equation (3) so that the supervised and regularization terms have approximately the same magnitude. All networks are trained using Adam [59] for 50 epochs, with a batch size of 60 and learning rate of 10^{-4}. Our trained networks are available at the project website [19].

Datasets. For our quantitative comparisons, we use a synthetic dataset containing images of translucent objects with varying geometry, illumination, and material parameters. We use ten different object shapes, selected to have a variety of thin and thick geometric features, each placed under ten different illumination conditions created using the Hošek-Wilkie sun-sky model [68]. For each shape and illumination combination, we render images for different parameters $\boldsymbol{\pi}$ that include $\sigma_t \in \left[25\,\mathrm{mm}^{-1}, 300\,\mathrm{mm}^{-1}\right]$,

$\alpha \in [0.3, 0.95]$, and Henyey-Greenstein phase functions f_p with parameter $g \in [0, 0.9]$. We use the Mitsuba physics-based renderer [61] to simulate 30,000 high-dynamic range images under these settings. This dataset is available at the project website [19].

We focus on evaluating the ability of the different networks to generalize to scenes containing new shapes and illuminations. For this, we separate our rendered images into training and testing sets that do not contain any overlapping shapes or illuminations. In particular, we use the images for six shapes and four illuminations as the training set, and use images under the remaining shape and illumination combinations for testing (Figure 4). This yields a training set of 6,000 images and a testing set of 7,000 images (we exclude a few thousand images available in the dataset that mix training illuminations with testing shapes, or vice versa). Our use of a training set that is relatively small compared to testing, and a testing set that contains only *completely unseen shapes and illuminations*, both reflect our goal to evaluate the generalization properties of the five networks we consider.

Finally, we note that, even though all networks are trained on grayscale images, they can handle color images by processing each color channel independently. Throughout this paper and in the supplement, we visualize results using color images, synthesized by combining grayscale images from our dataset that have the same illumintion and shape cnditions, but different material parameters. These color images are processed by the networks in the above-described channel-by-channel manner.



(a) shapes and illuminations used in training (left) and testing (right)

(b) network architecture

Fig. 4: **Datasets and architecture:** (a) We render 30,000 images for the task of homogeneous inverse scattering. These are split into training and testing sets of different shapes, illuminations, and materials. Rendered grayscale images under the same illumination and shape but with different material parameters are combined to form color images. (b) We use these datasets to train and evaluate regressor and inverse transport networks of the same architecture.

Quantitative Evaluation. We evaluate the five networks in two ways: First, we consider how accurately they predict material parameters for images in the testing dataset. Accuracy is quantified using the L_2 error between groundtruth and predicted parameters. Second, we examine how well images rendered with the predicted parameters match the appearance of the input images. We compare rendered and input images using L_2 error and MS-SSIM [69] (a benchmark perceptually-motivated image similarity metric).

TABLE 1: **Network performances:** Average MSE for individual scattering parameters, as well as L_2 and $1 - $ MS-SSIM image appearance errors, for five different networks.

network	parameters			appearance	
	σ_t	α	$\bar{c}$	L_2	1 - MS-SSIM
RN+naive	90.81	0.180	0.400	0.314	0.069
RN+similar	77.45	0.141	0.374	0.180	0.036
ITN+naive	71.23	0.093	**0.253**	0.222	0.031
ITN+similar	64.62	0.116	0.273	**0.144**	0.032
ITN+similar+map	**60.32**	**0.088**	0.282	0.145	**0.024**

Table 1 summarizes the results. The ITN with similarity-aware parameterization and weight maps outperforms the other three networks in most metrics, except for $\bar{c}$ where its performance is a close second. In terms of parameter prediction, it is noteworthy that the similarity-aware parameterization outperforms the naive pameterization for both the RN and ITN cases, despite the fact that errors are computed directly on the parameters optimized by the naive parameterization (σ_t, α, and $\bar{c}$). This highlights the importance of accounting for similarity relations when designing networks for inverse scattering. When comparing ITNs with RNs, the ITN produces strong improvements in both parameter and appearance predictions regardless of what parameterization is used. These improvements provide strong evidence that the regularization term in Equation (3) allows the ITN to generalize better to unseen shapes and illuminations. Finally, Figures 5 and 6 show images rendered with parameters predicted by the five networks, for materials of varying optical thickness, including diffusive, turbid, and very optically thin materials. In all cases, the ITN with similarity-aware parameterization and weight map produces the images that best match the reference.

Evaluation under novel scene. The previous results already focus on the generalization performance of the trained networks, considering that the testing images have shape and illumination conditions that are completely absent from the training dataset. To further emphasize generalization performance, we perform an additional set of experiments: We use the networks to predict material parameters for all test images. We then use these parameter estimates, as well as their groundtruth values, to render images for a scene of completely new geometry and lighting, absent from both training and testing datasets. We compare these renderings using the same image similarity metrics as above. Table 2 shows the results, and Figure 5 (rightmost column) visualizes a few example images rendered on this novel scene. We see that the ITN with similarity-aware parameterization and weight map significantly outperforms all other networks, and produces images very similar to those rendered with the groundtruth parameters. This provides evidence that this network can infer reliable estimates of the true scattering paremeters underlying a translucent object, from just a single, completely uncalibrated image of that object.

Initialization of analysis by synthesis. As we discussed in Section 4, our main focus is on using neural networks to predict scattering parameters from completely uncalibrated photographs, where analysis by synthesis optimization is not possible due to lack of information on geometry and illumination. When this information is available, analysis by synthesis will typically produce more accurate material parameters than our method, considering that analysis by synthesis uses significantly more information about the object. However, even in such cases, our networks can

Fig. 5: **Images rendered with predicted material parameters:** Each column corresponds to a different input image drawn from our synthetic testing set. The last column shows images rendered under the novel scene used to emphasize generalization performance. For each image, different rows compare the groundtruth (row 1) to images rendered using the parameters predicted by the five networks we evaluate (rows 2-6).

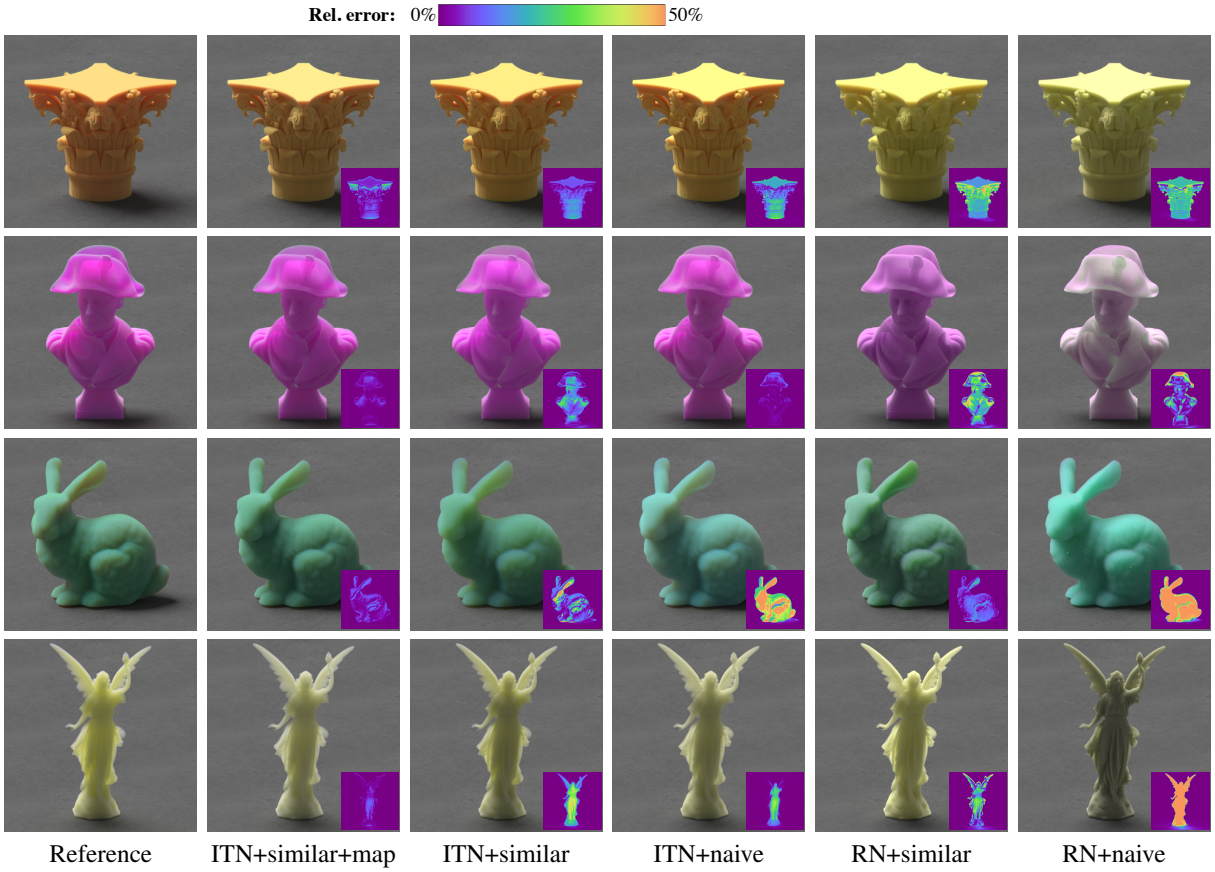


Fig. 6: **Images rendered with predicted material parameters:** Each row corresponds to a different input image drawn from our synthetic testing set. For each image, different columns compare the groundtruth (column 1) to images rendered using the parameters predicted by the five networks we evaluate (columns 2-6).

TABLE 2: **Network performances:** Average L_2 and $1 -$ MS-SSIM image appearance errors, for five different networks, on the novel scene used to emphasize generalization performance.

network	appearance	
	L_2	1 - MS-SSIM
RN+naive	0.250	0.067
RN+similar	0.144	0.035
ITN+naive	0.186	0.031
ITN+similar	0.140	0.035
ITN+similar+map	**0.133**	**0.021**

still be useful, as they provide a way to warm-start subsequent analysis by synthesis optimization, accelerating its convergence and improving the quality of the resulting material estimates.

To quantify the performance difference between our networks and analysis by synthesis, as the well as the advantage that can be gained from warm-starting, we perform the following experiment: We randomly select a subset of 40 images from the testing set (10 for each shape in that set), and use our five networks to predict material parameters. Then, we use these parameters to initialize analysis by synthesis optimization for each image: We use our differentiable renderer together with the groundtruth scene information (e.g., geometry and illumination) to compute derivatives of the loss of Equation (1) with respect to the parameters $\boldsymbol{\pi}$, and use ADAM to optimize the values of these parameters, starting from the values predicted by the networks. Each optimization procedure is run for 150 iterations, and we record the average MSE of individual parameters across all 40 images at every 30 iterations. The results are shown in Table 3. We make two observations: First, as expected, analysis by synthesis optimization takes advantage of the additional scene information it has access to, to significantly improve the initial parameter predictions of the networks (average MSE reduction of 43.8%). Second, initializing with our ITN with similarity-aware parameterization and weight map provides more than $2x$ convergence speedup compared to the baseline RN, resulting in 53.5% better average MSE at the same number of iteratons. Our ITN with similarity-aware parameterization and weight map additionally outperforms all other neworks.

The improvements observed above are in part due to the fact that a better initialization can help the analysis by synthesis procedure avoid local minima due to similarity relations. Figure 7 shows a simple demonstration: The ITN with similarity-aware parameterization and weight map produces an initialization that is closer to the true parameters than the one by the ITN with naive parameterization. Additionally, after 50 gradient descent iterations, the optimization initialized by the ITN with similarity-aware parameterization and weight map has converged to parameters closer to the groundtruth, also producing 7.7% lower image error. From the optimization trajectories at the top of Figure 7, we observe that, after a few iterations, the optimization initialized by the ITN with naive parameterization moves along a contour of constant σ_s^r (visualized by the color map). This indicates that it

TABLE 3: **Network performances on initialization of analysis by synthesis:** Average MSE for individual scattering parameters over iterations for five different networks.

		number of iterations					
		1	30	60	90	120	150
σ_t	RN+naive	90.40	72.87	60.93	55.58	52.22	49.26
	RN+similar	59.23	43.96	34.36	28.87	25.41	23.67
	ITN+naive	60.02	57.82	51.36	47.27	43.69	40.51
	ITN+similar	50.45	39.47	33.09	28.78	25.53	23.27
	ITN+similar+map	**43.98**	**32.42**	**26.08**	**23.71**	**22.48**	**21.36**
α	RN+naive	0.138	0.095	0.096	0.093	0.089	0.084
	RN+similar	0.078	0.058	0.053	0.049	0.046	0.044
	ITN+naive	0.102	0.090	0.084	0.077	0.071	0.066
	ITN+similar	0.073	0.057	0.051	0.046	0.043	0.040
	ITN+similar+map	**0.057**	**0.052**	**0.047**	**0.043**	**0.041**	**0.038**
$\bar{c}$	RN+naive	0.304	0.279	0.266	0.268	0.264	0.252
	RN+similar	**0.232**	0.229	0.202	0.178	0.158	0.145
	ITN+naive	0.284	0.246	0.224	0.207	0.190	0.178
	ITN+similar	0.233	0.234	0.210	0.178	0.158	0.143
	ITN+similar+map	0.240	**0.202**	**0.172**	**0.151**	**0.139**	**0.128**

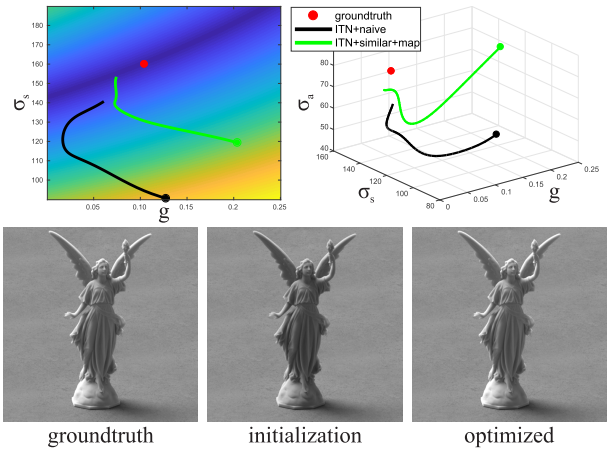


Fig. 7: **Post-learning refinement:** We use predictions from two ITNs to initialize analysis by synthesis optimization. The top row shows how parameters change during gradient descent, in the $\{\sigma_s, \sigma_a, \bar{c}\}$ parameter space, and in its 2D projection $\{\sigma_s, \bar{c}\}$ colored by reduced scattering coefficient value σ_s'. The bottom row compares groundtruth to renderings using the initial and final parameter estimations from the green optimization trajectory.

may be trapped at a local minimum. The bottom row of Figure 7 compares the groundtruth image with renderings using the initial and optimized material predictions from the ITN with similarity-aware parameterization and weight map.

Experiments on real photographs. Figure 8 shows results from using our top-performing networks, the three ITNs, on photographs of two translucent objects: a silicone cube, and a soap bar. The networks take as input a single high-dynamic-range photograph of the object, with completely uncalibrated geometry and illumination. To evaluate the network predictions, we created virtual scenes that crudely approximate the shape (by fitting rectangles) and lighting conditions (by matching shadows) of the original photographs. We then used these scenes to render images with the predicted parameters. We emphasize that these approximate scene conditions are used for validation only, and they are not used by the networks when making predictions.

We observe that, even though the renderings do not reproduce the appearance of the original objects perfectly, in all cases the parameter predictions produce plausible appearance, especially considering the complete lack of calibration. The appearance errors are in part because of the mismatched geometry and lighting, and the fact that we do not model surface reflectance. In particular, the rendered images reproduce important features of translucent appearance, e.g., the intensity gradients near geometric edges. Qualitatively, the ITN with similarity-aware parameterization and weight map performs the best among the three networks, in terms of both matching the overall intensity of the real objects, and intensity gradients at geometric edges. We consider these results a promising start towards uncalibrated and computationally efficient inverse subsurface scattering on images captured in-the-wild.

7 Conclusions

We have taken first steps towards using deep learning techniques for the problem of homogeneous inverse scattering. Starting from a state-of-the-art regression network architecture as baseline, we made three innovations, informed from the physics of radiative transfer: First, we introduced inverse transport networks as an architecture that can combine the efficiency of neural networks with the generality properties of analysis by synthesis. Second, we used material parameterizations that can ameliorate ambiguities in the scattering parameter space due to similarity relations. Third, we utilized per-pixel weight maps to emphasize parts of the image that are informative about the underlying scattering parameters. Our experiments show that the combination of these innovations results in networks that can produce convincing scattering material parameter estimates, when provided with a single photograph without any calibration (completely unknown geometry and illumination). Additionally, the performance of our networks shows strong improvements in both parameter estimation accuracy and appearance reproduction compared to the baseline. We hope that these results will motivate follow-up work on using data-driven learning techniques to improve upon and complement existing physics-based approaches for inverse subsurface scattering.

At the core of our approach is the use of Monte Carlo differentiable renderers. The use of physically-accurate rendering allows us to enhance the generalization of neural networks, and the differentiability allows us to efficiently train these networks. Together with other work on combining differentiable rendering with learning [16], our results point towards a new direction of exploration: the investigation of more general learning architectures that intelligently combine neural networks with physics-based light transport simulation. We hope that our paper and our publicly-available implementations and datasets [19] will facilitate further research in this direction.

Acknowledgments

This work was supported by NSF Expeditions award 1730147, NSF awards IIS-1900783, IIS-1900849, and IIS-1900927, and a gift from the AWS Cloud Credits for Research program.

References

[1] J. Novák, I. Georgiev, J. Hanika, J. Křivánek, and W. Jarosz, "Monte carlo methods for physically based volume rendering," *SIGGRAPH Courses*, 2018.

[2] S. Narasimhan, M. Gupta, C. Donner, R. Ramamoorthi, S. Nayar, and H. Jensen, "Acquiring scattering properties of participating media by dilution," *ACM TOG*, 2006.

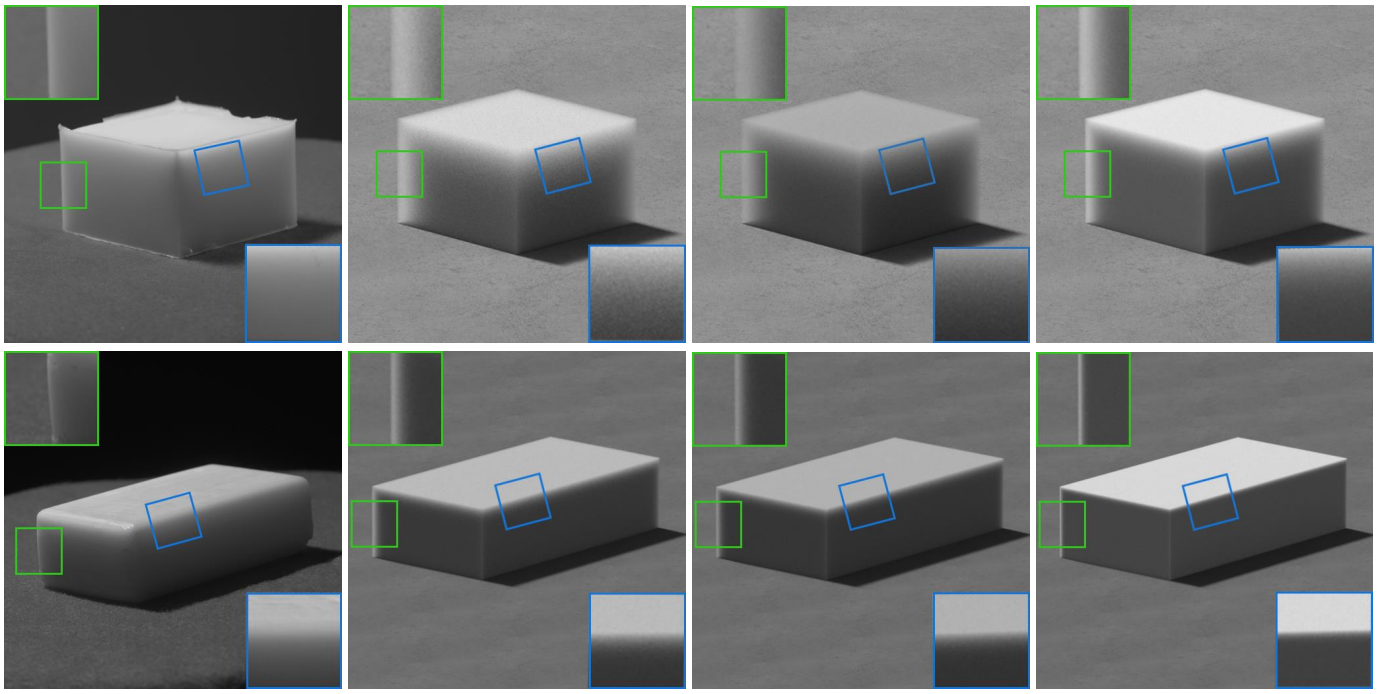


Fig. 8: **Real photographs:** We take photographs of two translucent objects, a silicone cube (top) and a soap bar (bottom), under unknown shape and illumination (left column). We use three ITNs to predict scattering parameters for the objects. We evaluate the predicted parameters by rendering images for an approximate reconstruction of the original scenes (middle and right column).

[3] H. Jensen, S. Marschner, M. Levoy, and P. Hanrahan, "A practical model for subsurface light transport," 2001.

[4] I. Gkioulekas, S. Zhao, K. Bala, T. Zickler, and A. Levin, "Inverse volume rendering with material dictionaries," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 162:1–162:13, 2013.

[5] S. Zhao, L. Wu, F. Durand, and R. Ramamoorthi, "Downsampling scattering parameters for rendering anisotropic media," *ACM TOG*, 2016.

[6] P. Khungurn, D. Schroeder, S. Zhao, K. Bala, and S. Marschner, "Matching real fabrics with micro-appearance models," *ACM TOG*, 2015.

[7] I. Gkioulekas, A. Levin, and T. Zickler, "An evaluation of computational imaging techniques for heterogeneous inverse scattering," *ECCV*, 2016.

[8] A. Levis, Y. Schechner, A. Aides, and A. Davis, "Airborne three-dimensional cloud tomography," *ICCV*, 2015.

[9] S. R. Marschner and D. P. Greenberg, *Inverse rendering for computer graphics*. Cornell University, 1998.

[10] T. Weyrich, J. Lawrence, H. P. Lensch, S. Rusinkiewicz, T. Zickler *et al.*, "Principles of appearance acquisition and representation," *Foundations and Trends® in Computer Graphics and Vision*, 2009.

[11] A. Meka, M. Maximov, M. Zollhöfer, H.-P. Seidel, C. Richardt, and C. Theobalt, "Lime: Live intrinsic material estimation," *CVPR*, 2018.

[12] S. Sengupta, A. Kanazawa, C. D. Castillo, and D. W. Jacobs, "Sfsnet: Learning shape, reflectance and illuminance of faces 'in the wild'," *ICCV*, 2018.

[13] G. Liu, D. Ceylan, E. Yumer, J. Yang, and J.-M. Lien, "Material editing using a physically based rendering network," *ICCV*, 2017.

[14] H. Kato, Y. Ushiku, and T. Harada, "Neural 3d mesh renderer," *CVPR*, 2018.

[15] A. Tewari, M. Zollhöfer, H. Kim, P. Garrido, F. Bernard, P. Pérez, and C. Theobalt, "Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction," *ICCV*, 2017.

[16] T.-M. Li, M. Aittala, F. Durand, and J. Lehtinen, "Differentiable monte carlo ray tracing through edge sampling," *ACM TOG*, 2018.

[17] D. Wyman, M. Patterson, and B. Wilson, "Similarity relations for the interaction parameters in radiation transport," *Applied optics*, 1989.

[18] S. Zhao, R. Ramamoorthi, and K. Bala, "High-order similarity relations in radiative transfer," *ACM TOG*, 2014.

[19] "Project website," 2020, https://imaging.cs.cmu.edu/inverse_transport_networks.

[20] M. Mishchenko, L. Travis, and A. Lacis, *Multiple scattering of light by particles: radiative transfer and coherent backscattering*. Cambridge University, 2006.

[21] T. Hawkins, P. Einarsson, and P. Debevec, "Acquisition of time-varying participating media," *ACM TOG*, 2005.

[22] C. Fuchs, T. Chen, M. Goesele, H. Theisel, and H. Seidel, "Density estimation for dynamic volumes," *Computers & Graphics*, 2007.

[23] J. Gu, S. Nayar, E. Grinspun, P. Belhumeur, and R. Ramamoorthi, "Compressive structured light for recovering inhomogeneous participating media," *ECCV*, 2008.

[24] J. Wang, S. Zhao, X. Tong, S. Lin, Z. Lin, Y. Dong, B. Guo, and H. Shum, "Modeling and rendering of heterogeneous translucent materials using the diffusion equation," *ACM TOG*, 2008.

[25] M. Papas, C. Regg, W. Jarosz, B. Bickel, P. Jackson, W. Matusik, S. Marschner, and M. Gross, "Fabricating translucent materials using continuous pigment mixtures," *ACM TOG*, 2013.

[26] B. Dong, K. D. Moore, W. Zhang, and P. Peers, "Scattering parameters and surface normals from homogeneous translucent materials using photometric stereo," *CVPR*, 2014.

[27] C. Inoshita, Y. Mukaigawa, Y. Matsushita, and Y. Yagi, "Surface normal deconvolution: Photometric stereo for optically thick translucent objects," *ECCV*, 2014.

[28] A. Munoz, J. I. Echevarria, F. Seron, J. Lopez-Moreno, M. Glencross, and D. Gutierrez, "BSSRDF estimation from single images," *CGF*, 2011.

[29] Y. Song, X. Tong, F. Pellacini, and P. Peers, "Subedit: A representation for editing measured heterogeneous subsurface scattering," *ACM TOG*, 2009.

[30] Y. Mukaigawa, K. Suzuki, and Y. Yagi, "Analysis of subsurface scattering based on dipole approximation," *IPSJ TCVA*, 2009.

[31] S. Kallweit, T. Müller, B. McWilliams, M. Gross, and J. Novák, "Deep scattering: Rendering atmospheric clouds with radiance-predicting neural networks," *ACM TOG*, 2017.

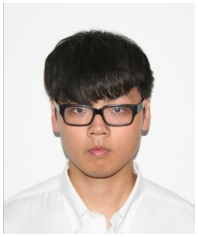
[32] D. Vicini, V. Koltun, and W. Jakob, "A learned shape-adaptive subsurface scattering model," *ACM TOG*, 2019.

[33] P. Gargallo, E. Prados, and P. Sturm, "Minimizing the reprojection error in surface reconstruction from images," *CVPR*, 2007.

[34] A. Delaunoy and E. Prados, "Gradient flows for optimizing triangular mesh-based surfaces: Applications to 3d reconstruction problems dealing with visibility," *IJCV*, 2011.

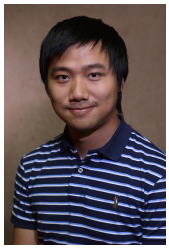
[35] F. Romeiro and T. Zickler, "Blind reflectometry," *ECCV*, 2010.

[36] Y. Mukaigawa, Y. Yagi, and R. Raskar, "Analysis of light transport in scattering media," *CVPR*, 2010.
[37] K. Nishino, "Directional statistics brdf model," *ICCV*, 2009.
[38] X. Mei, H. Ling, and D. W. Jacobs, "Illumination recovery from image with cast shadows via sparse representation," *IEEE TIP*, 2011.
[39] J. T. Barron and J. Malik, "Shape, illumination, and reflectance from shading," *IEEE TPAMI*, 2015.
[40] K.-J. Yoon, E. Prados, and P. Sturm, "Joint estimation of shape and reflectance using multiple images with known illumination conditions," *IJCV*, 2010.
[41] S. Lombardi and K. Nishino, "Reflectance and illumination recovery in the wild," *IEEE TPAMI*, 2016.
[42] G. Oxholm and K. Nishino, "Multiview shape and reflectance from natural illumination," in *CVPR*, 2014.
[43] M. M. Loper and M. J. Black, "Opendr: An approximate differentiable renderer," *ECCV*, 2014.
[44] S. Lombardi and K. Nishino, "Radiometric scene decomposition: Scene reflectance, illumination, and geometry from rgb-d images," *IEEE 3DV*, 2016.
[45] D. Azinovic, T.-M. Li, A. Kaplanyan, and M. Niessner, "Inverse path tracing for joint material and lighting estimation," *CVPR*, 2019.
[46] C.-Y. Tsai, A. Sankaranarayanan, and I. Gkioulekas, "Beyond volumetric albedo—a surface optimization framework for non-line-of-sight imaging," *CVPR*, 2019.
[47] C. Zhang, L. Wu, C. Zheng, I. Gkioulekas, R. Ramamoorthi, and S. Zhao, "A differential theory of radiative transfer," *ACM TOG*, 2019.
[48] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob, "Mitsuba 2: A retargetable forward and inverse renderer," *ACM TOG*, 2019.
[49] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *JMLR*, 2010.
[50] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *ICLR*, 2014.
[51] J. Wu, J. B. Tenenbaum, and P. Kohli, "Neural scene de-rendering," *CVPR*, 2017.
[52] V. Deschaintre, M. Aittala, F. Durand, G. Drettakis, and A. Bousseau, "Single-image svbrdf capture with a rendering-aware deep network," *ACM TOG*, 2018.
[53] Z. Shu, E. Yumer, S. Hadap, K. Sunkavalli, E. Shechtman, and D. Samaras, "Neural face editing with intrinsic image disentangling," *CVPR*, 2017.
[54] M. Aittala, T. Aila, and J. Lehtinen, "Reflectance modeling by neural texture synthesis," *ACM TOG*, 2016.
[55] K. Genova, F. Cole, A. Maschinot, A. Sarna, D. Vlasic, and W. T. Freeman, "Unsupervised training for 3d morphable model regression," *CVPR*, 2018.
[56] Z. Li, K. Sunkavalli, and M. Chandraker, "Materials for masses: Svbrdf acquisition with a single mobile phone image," *ECCV*, 2018.
[57] T. Nguyen-Phuoc, C. Li, S. Balaban, and Y. Yang, "A deep convolutional network for differentiable rendering from 3d shapes," *NeurIPS*, 2018.
[58] S. Sengupta, J. Gu, K. Kim, G. Liu, D. W. Jacobs, and J. Kautz, "Neural inverse rendering of an indoor scene from a single image," *ICCV*, 2019.
[59] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, 2015.
[60] B. Carpenter, M. D. Hoffman, M. Brubaker, D. Lee, P. Li, and M. Betancourt, "The Stan Math Library: Reverse-mode automatic differentiation in C++," *arXiv preprint arXiv:1509.07164*, 2015.
[61] W. Jakob, "Mitsuba renderer," 2010, http://www.mitsuba-renderer.org.
[62] L. Bottou and O. Bousquet, "The Tradeoffs of Large Scale Learning," *NeurIPS*, 2008.
[63] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *NeurIPS*, 2012.
[64] I. Gkioulekas, B. Walter, E. Adelson, K. Bala, and T. Zickler, "On the appearance of translucent edges," *CVPR*, 2015.
[65] R. W. Fleming and H. H. Bülthoff, "Low-level image cues in the perception of translucent materials," *ACM TAP*, 2005.
[66] I. Gkioulekas, B. Xiao, S. Zhao, E. Adelson, T. Zickler, and K. Bala, "Understanding the role of phase function in translucent appearance," *ACM TOG*, 2013.
[67] N. S. Chowdhury, P. J. Marlow, and J. Kim, "Translucency and the perception of shape," *JOV*, 2017.
[68] L. Hosek and A. Wilkie, "An analytic model for full spectral sky-dome radiance," *ACM TOG*, 2012.
[69] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," *Asilomar CSSC*, 2003.



Chengqian Che is a Ph.D student in the Robotics Institute at Carnegie Mellon University. He is currently working with Professor Ioannis Gkioulekas on inverse rendering especially for sub-surface scattering material. Before that, he obtained his Master's degree in Biomedical Engineering at CMU, focusing on medical image analysis, advised by Professor John Galeotti and Professor George Stetten.

Fujun Luan is a PhD student in the Department of Computer Science, Cornell University. Before that, he received his bachelor degree from Tsinghua University in 2015. His research interests are in physically-based rendering, image editing and neural networks.



Shuang Zhao is an Assistant Professor in the Department of Computer Science at University of California, Irvine and co-direct UCI's Interactive Graphics and Visualization Lab (iGravi). Before joining UCI, he was a postdoctoral associate at MIT. Shuang received his M.S. and Ph.D. in computer science from Cornell University. His research focuses on material appearance modeling, physics-based rendering, and material perception.

Kavita Bala is the Chair of the Computer Science Department at Cornell University. She received her BTech degree from the Indian Institute of Technology, Bombay, and her SM and PhD degrees from the Massachusetts Institute of Technology. She specializes in computer vision and computer graphics, leading research in recognition and visual search; material modeling and acquisition; physically-based rendering; and material perception. She was the editor-in-chief of the ACM Transactions on Graphics (TOG, 2015–2018). She has also served on the Papers Advisory Group for SIGGRAPH, and as associate editor for IEEE Transactions on Visualization and Computer Graphics, and the Computer Graphics Forum. She has co-authored the graduate-level textbook "Advanced Global Illumination". She has chaired SIGGRAPH Asia 2011, and co-chaired Pacific Graphics (2010) and the Eurographics Symposium on Rendering (2005). She is an ACM Fellow, and a member of the IEEE.

Ioannis Gkioulekas is an Assistant Professor at the Robotics Institute, Carnegie Mellon University. He received M.Sc. and Ph.D. degrees from Harvard University, and a Diploma of Engineering from the National Technical University of Athens. His research interests are on computational imaging, physics-based rendering, and differentiable rendering. He has received the Best Paper Award at CVPR 2019, and a Sloan Research Fellowship.