L-Tracing: Fast Light Visibility Estimation on Neural Surfaces by Sphere Tracing

Ziyu Chen[1†], Chenjing Ding[2,3], Jianfei Guo[3], Dongliang Wang[2,3], Yikang Li[2,3], Xuan Xiao[4], Wei Wu[2,3], and Li Song[1‡]

[1] Department of Electronic Engineering, Shanghai Jiao Tong University
[2] SenseTime Research, {dingchenjing,wangdongliang,wuwei}@senseauto.com
[3] Shanghai AI Laboratory, {guojianfei,liyikang}@pjlab.org.cn
[4] Department of Mechanical Engineering, Tsinghua University
x-xiao20@mails.tsinghua.edu.cn

Abstract. We introduce a highly efficient light visibility estimation method, called L-Tracing, for reflectance factorization on neural implicit surfaces. Light visibility is indispensable for modeling shadows and specular of high quality on object's surface. For neural implicit representations, former methods of computing light visibility suffer from efficiency and quality drawbacks. L-Tracing leverages the distance meaning of the Signed Distance Function(SDF), and computes the light visibility of the solid object surface according to binary geometry occlusions. We prove the linear convergence of L-Tracing algorithm and give out the theoretical lower bound of tracing iteration. Based on L-Tracing, we propose a new surface reconstruction and reflectance factorization framework. Experiments show our framework performs nearly ~10x speedup on factorization, and achieves competitive albedo and relighting results with existing approaches.

Keywords: surface reflectance factorization, sphere tracing, inverse rendering, 3D deep learning

1 Introduction

Reconstructing an object's surface, reflectance property, and environment illumination from multi-view images is a fundamental problem in computer vision and graphics. Deep learning methods for objects' reconstruction save human resources for producing high-quality 3D models. Recent works [29, 24, 33] adopt neural volume rendering to learn detailed surfaces of high quality from multi-view images. Researchers [2, 5, 36, 22, 28, 39, 26, 3, 32] further explore the field that decomposes the reflectance of neural surfaces and enviornment illumination, to produce renderable 3D models with de-illuminated textures.

In the area of surface reflectance and ambient illumination decomposition, estimating whether the surface point is visible to light sources is called light

[†] The work is done when Ziyu Chen <ziyu.sjtu@gmail.com> is an intern at SenseTime.
[‡] Corresponding author: Li Song. <song_li@sjtu.edu.cn>

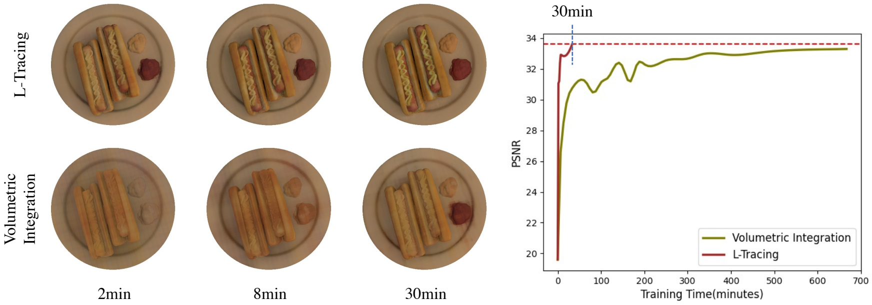


Fig. 1. L-Tracing is a highly efficient light visibility estimation method for reflectance factorization on neural implicit surfaces. We show that our L-Tracing based reflectance factorization framework adapted from NeRFactor [39] produces photo-realistic novel view images with nearly 10x speedup, compared with the same framework applying volumetric integration for light visibility estimation.

visibility estimation. Since the color of each observed pixel is aggregated by the reflected rays on surface from light sources, light visibility is important for modeling shadows and the specular effect.

Recently, there are several works studying reflectance factorization of neural volume representation [26, 39, 3, 37]. The light visibility in these studies is regarded as the possibility the light passing to a certain point. Under this assumption, the differential probability distribution along each ray is needed to compute light visibility by volumetric integration. This is computationally expensive since hundreds of points on each ray are sampled to compute the discrete form of volumetric integration.

NeRFactor [39] and NeRV [26] reduce the computational cost by estimating light visibility via a Multi Layer Perceptron(MLP), which is trained by precomputed light visibility caches. These methods only reduce the computational cost of inference time, the caching process still depends on volumetric integration, consuming hundreds of hours which buries a heavy burden at the training time.

We note that previous work performing reflectance decomposition on neural implicit shapes suffer from the following drawbacks: 1)Some methods [32, 37, 3, 12] fail to explicitly model the light visibility, where the learning process entirely depends on neural networks to learn the geometry occlusion and shadows. 2)Some methods [39, 26] show poor computation efficiency because of using the volumetric integration to compute the light visibility. In this way, computing the light visibility once usually queries hundreds of points along the ray for the density. 3)Under the differential probability assumption of neural volume representation, some methods [39, 26, 3] incorrectly model the objects as translucent volumes instead of solid spatial boundaries, producing low-quality surfaces.

Our goal is to present a highly efficient and accurate light visibility estimation method that explicitly respects solid objects and binary geometry occlusions. We make up the third drawback by using Signed Distance Function(SDF) as our shape representation. SDF is a continuous implicit surface representation that is easily optimized by gradient decent. The signed distance induces a highly efficient ray tracing algorithm: sphere tracing. We propose L-tracing based on sphere tracing, to solve the computational problem for light visibility estimation. Different from the former assumption, the light visibility in our work is estimated according to hard spatial occlusions instead of integrated visible probability, which improves the learning of sharp shadows and specular effects. Moreover, L-tracing achieves ~10x faster than volumetric integration. We also provide the convergence proof of L-tracing.

We propose a two-stage learning framework for reconstructing objects' surface and decomposing the reflectance and environment illumination. First, we adopt recent works [29] of reconstructing SDF from multi-view images without object masks via differentiable volume rendering. Second, we apply L-Tracing as our light visibility estimation method, and jointly optimize the albedo, reflectance and environment illumination in a similar manner to [39]. As shown in Fig. 1, we compare the convergence speed and novel views' quality of the reflectance factorization stage based on L-Tracing against those based on volumetric integration. The results show that L-Tracing is far more efficient than volumetric integration.

In summary, our work makes the following contributions :

• We propose L-tracing, a highly efficient light visibility estimation method on neural implicit surfaces, to accelerate the rendering process for reflectance decomposition tasks. We also prove the convergence of L-Tracing theoretically.

• We propose a two-stage framework for neural implicit surface reconstruction and reflectance factorization using the proposed L-Tracing as the light visibility estimation method.

• L-Tracing is ~10x faster than estimating light visibility based on volumetric integration. The rendered novel views and relighting images from the reconstructed object surface and reflectance achieve comparable quality with recent works across different metrics.

2 Related Works

2.1 Neural Shape Representation

Neural shape representations are widely used in 3D shape reconstruction and inverse rendering tasks. NeRF [21] and its variants [10, 8, 17, 35, 20, 30, 38, 23, 27, 9] use volume differentiable rendering on volumetric representation achieving good quality on novel view synthesis. Although they have achieved great success in the field of free-view interpolation, they struggle at recovering solid object boundaries (object surfaces) due to the lack of surface definition of volume representation.

Recent works [25, 19, 34, 6] studying 3D surface reconstruction tend to use SDF as learnable shape representation. SDF is a function of spatial coordinates, with its value indicating the minimum distance between the input coordinate and the surface. Because of the simple and continuous parameterization, SDF is suitable to learn well-shaped neural surfaces. [29, 33, 24] explored the way to reconstruct an implicit surface by combining the advantages of both volume rendering and surface rendering. In our work, we use SDF as our neural implicit shape representation, and adopt [29] as our first stage of reconstructing well-shaped surfaces from multi-view unmasked images. We then apply L-tracing on trained neural surfaces to decompose the reflectance and material from unknown ambient illumination.

2.2 Reflectance and illumination estimation

Decomposing the object surface into materials, reflectance, and ambient illumination from multi-view images is one of the main concerns of the inverse rendering tasks. Since this is a highly ill-posed problem, traditional works [4, 14, 15, 1, 16] often rely on priors.

Recent works tend to use the neural rendering method [2, 5, 36, 22, 28, 39, 26, 3, 32, 12, 16, 37, 13, 31] to learn the object BRDF and environment illumination from image collections. NeRD [3] decompose reflectance from images under different illumination. PhySG [37] models geometry by SDF and the environment illumination by the mixtures of spherical Gaussians. Neural Ray-Tracing [12] leverages sphere tracing to model multi-bounce reflection. NeRD [3], PhySG [37] and Neural Ray-Tracing [12] don't model the light visibility, the intensity of the incoming light is learned totally by deep networks. NeRV [26] and NeRFactor [39] model the light visibility by volumetric integration of the density along the ray and train a MLP for light visibility prediction. However, learning from the network loses details of high frequency topology, which will be discussed in the experiment section. Instead of doing reflectance factorization on neural representations, [22] jointly optimize the explicit geometry, material and illumination, the images are rendered by a differentiable rasterizer with deferred shading.

As demonstrated in Section 1, the works mentioned above either fail to model light visibility physically, or do modeling at an expensive computation cost by volumetric integration. Our proposed L-Tracing models the direct illumination of the environment, estimating the light visibility based on physical geometry occlusions, accelerates the learning process for reflectance factorization.

3 Light Visibility Estimation

3.1 Preliminaries

We first review the volume rendering process that is developing rapidly in the area of novel view synthesis. Consider rendering a ray starting from $\boldsymbol{o}$ and in the normalized direction of $\boldsymbol{d}$. The points on this ray is noted by $\{\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d} \mid t \geq 0\}$. The color of the pixel corresponding to this ray is:

$$L(\boldsymbol{o}, \boldsymbol{d}) = \int_0^{+\infty} w(t) c(\boldsymbol{r}(t), \boldsymbol{d}) \mathrm{d}t \tag{1}$$

where $w(t)$ denotes the ratio of each sampled point's color $c(\boldsymbol{r}(t), \boldsymbol{d})$ contributing to the corresponding pixel color $L(\boldsymbol{o}, \boldsymbol{d})$. $w(t)$ is further expanded into

$$w(t) = T(t)\sigma(\boldsymbol{r}(t)), \quad \text{where } T(t) = \exp\left(-\int_0^t \sigma(\boldsymbol{r}(u))\mathrm{d}u\right) \tag{2}$$

$T(t)$ denotes the probability of the ray being transmitted to $\boldsymbol{r}(t)$, and $\sigma(\boldsymbol{r}(t))$ denotes the differential probability the ray being omitted at the position $\boldsymbol{r}(t)$.

The light visibility of spatial point is defined by the accumulated transmittance on the path. Fig. 2(b) demonstrates the light visibility of the volumetric integration process. Consider a ray emitted from the light source noted by $\{\boldsymbol{r}_l(t) = \boldsymbol{x}_l + t\boldsymbol{d}_l \mid t \geq 0\}$, and the spatial point along the ray noted by $\boldsymbol{x}_\mathrm{s} = \boldsymbol{r}_l(t_s)$. The light visibility v based on volumetric integration is:

$$v(\boldsymbol{x}_l, \boldsymbol{x}_\mathrm{s}) = 1 - \int_0^{t_s} T(t)\sigma(\boldsymbol{r}_l(t))\mathrm{d}t \tag{3}$$

The light visibility computed by volumetric integration models the probability of the light being blocked along the way, which is a continuous value ranging from 0 to 1. Hence, the volumetric integration process is considered physically incorrect as for direct illumination and binary geometry occlusions, because in most of the real-world cases objects are typically solid and not translucent. Motivated by this, we introduce L-Tracing, a highly efficient light visibility estimation method for neural implicit surfaces that respect solid objects and hard geometry occlusions.

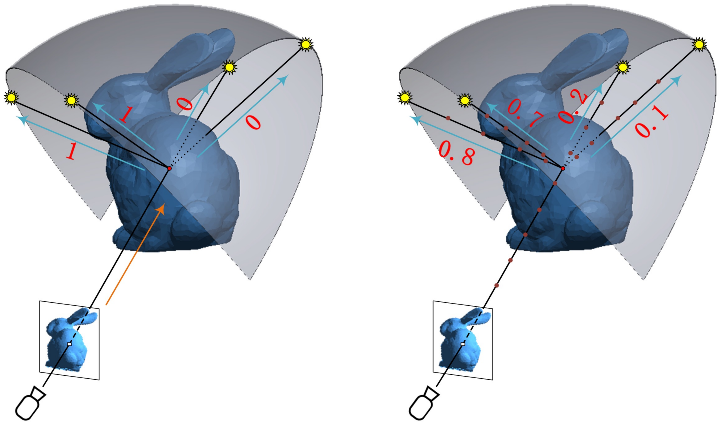
3.2 L-Tracing

To alleviate the physical incorrectness of volume rendering's light visibility and accelerate the process of computing light visibility, we propose L-Tracing, adopting sphere tracing for light visibility estimation on neural surfaces encoded by learned SDF. L-Tracing computes whether the light path is occluded by a surface leveraging the signed distance output.

Denoting the signed distance function as f, the neural implicit surface is defined as the zero-level set of f:

$$\mathcal{S} = \left\{\boldsymbol{x} \in \mathbb{R}^3 \mid f(\boldsymbol{x}) = 0\right\} \tag{4}$$

The light visibility v from light source $\boldsymbol{x}_l$ to surface point $\boldsymbol{x}_s \in \mathcal{S}$ is computed following Algorithm 1. We first calculate the upper bound of the tracing range, that is the euclidean distance between $\boldsymbol{x}_l$ and $\boldsymbol{x}_s$. At the start of the iteration, a tracing point on light position is queried for its signed distance (sd for short), to decide the signed distance step for the next iteration. If the tracing point moves beyond the range, the returned light visibility v is 1, indicating there is no surface that occludes the light path. If the tracing point is trapped after T iterations, then return light visibility $v = 0$, which means the light is blocked by the surface.



(a) L-Tracing (b) Volumetric Integration

Fig. 2. Illustration of Light Visibility Computation. (a) shows binary visibility computed by L-Tracing, the surface point is invisible to the right lights because of the solid geometry occlusion, thus the light visibility is 0. (b) shows the continuous visibility computed by volumetric integration. Although the surface point is invisible to the right lights, volumetric integration still computes the number greater than 0.

According to the assumption of L-Tracing, light visibility models binary geometry occlusions on the solid object surface, which helps better produce photo-realistic shadows and specular effects. This will be discussed in Section 5.1. In our work under the direct-illumination-only setting, light visibility is represented by the binary value in $\{0, 1\}$ which represents the binary geometry occlusion, instead of the continuous value ranging from 0 to 1 that derived from volumetric integration. Besides, L-Tracing exhibits good performance on computational efficiency, because of the linear convergence, enabling computing light visibility in fewer iterations with very high accuracy.

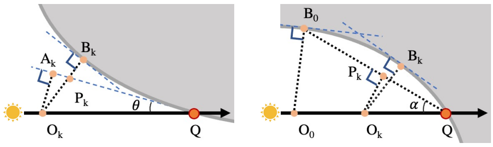
Convergence Proof. We prove that the high computational efficiency of L-Tracing comes from the linear convergence of sphere tracing. In Fig. 3 we show two surface conditions of L-Tracing. Among the Figure, $\boldsymbol{O}_k = \boldsymbol{x}_l + t_k\boldsymbol{d}_l$ denotes the tracing point at k_{th} iteration, $\boldsymbol{Q}$ denotes the observed surface point, $\boldsymbol{B}_k$ is the nearest surface points of $\boldsymbol{O}_k$. The distance between $\boldsymbol{Q}$ and $\boldsymbol{O}_k$ is ϵ_k. f_k is the signed distance of $\boldsymbol{O}_k$.

When the surface is convex, as shown in Fig. 3(a). We have $|\boldsymbol{O}_k\boldsymbol{B}_k| = f_k$, $|\boldsymbol{O}_k\boldsymbol{A}_k| = \epsilon_k sin\theta$, and $|\boldsymbol{O}_k\boldsymbol{A}_k| \leq |\boldsymbol{O}_k\boldsymbol{P}_k| \leq |\boldsymbol{O}_k\boldsymbol{B}_k|$ (the equal sign should be assigned when the surface can be represented as a local plane), from them we derive:

$$\epsilon_k sin\theta \leq f_k \tag{5}$$

Algorithm 1 L-Tracing

```
Input: $\boldsymbol{x}_l, \boldsymbol{x}_s, f$
Output: $v$
r = $\boldsymbol{x}_s - \boldsymbol{x}_l$
far = $\|\boldsymbol{r}\|_2$
$\boldsymbol{d} = \boldsymbol{r}/\|\boldsymbol{r}\|_2$
$t = 0$
for $i$ in range($T$) do
    pts = $\boldsymbol{x}_l + \boldsymbol{d} * t$
    sd = $f$(pts)
    $t = t +$ sd
    if $t >$ far or $t < 0$ then
        return 1
    end if
end for
return 0
```



(a) Convex Surface (b) Concave Surface

Fig. 3. Illustration of L-Tracing. We consider the ray of the light source is encountered with (a) a convex surface, (b) a concave surface. Since the plane surface can be regarded as the special case of (b), we don't list it separately for discussion.

As the tracing point in $(k+1)_{th}$ and k_{th} iteration satisfy $\epsilon_{k+1} = \epsilon_k - f_k$, from Eqn. 5 we have $\epsilon_{k+1} \leq \epsilon_k - \epsilon_k sin\theta$. Since ϵ is monotonic and bounded during iterations(Please refer to the supplementary material for the Proof), $\epsilon_k > 0$ is always satisfied. Then we get the following inequality:

$$0 < \frac{\epsilon_{k+1}}{\epsilon_k} \leq 1 - sin\theta \tag{6}$$

As for convex surface, we calculate the upper bound of the ratio between ϵ_{T-1} and ϵ_0:

$$0 < \frac{\epsilon_{T-1}}{\epsilon_0} \leq (1 - sin\theta)^T \leq \frac{\epsilon}{\epsilon_0} \tag{7}$$

where ϵ is the accuracy of convergence. Since $\theta \in (0, 90°)$, after a transformation, we have:

$$T \geq \frac{log\frac{\epsilon}{\epsilon_0}}{log(1 - sin\theta)} \tag{8}$$

When the surface is concave, as shown in Fig. 3(b). We have $\angle \boldsymbol{B}_0\boldsymbol{Q}\boldsymbol{O}_0 = \alpha$; $|\boldsymbol{O}_k\boldsymbol{B}_k| = f_k$; $\boldsymbol{O}_k\boldsymbol{P}_k \perp \boldsymbol{B}_0\boldsymbol{Q}$; Since $|\boldsymbol{O}_k\boldsymbol{P}_k| < |\boldsymbol{O}_k\boldsymbol{B}_k|$ is always satisfied, we get $\epsilon_k sin\alpha = |\boldsymbol{O}_k\boldsymbol{P}_k| \leq f_k$, from Eqn. 6, the following inequality is derived:

$$0 < \frac{\epsilon_{T-1}}{\epsilon_0} \leq (1 - sin\alpha)^T \leq \frac{\epsilon}{\epsilon_0} \tag{9}$$

According Eqn. 9, we get the theoretical minimum iterations needed for achieving the accuracy $\epsilon = 0.001$. We assume a ray intersects surface with an angle less than 5 degrees can be regarded as approximately parallel to the surface, thus, θ_{min} and α_{min} can be set to 5 degrees. The neural implicit surface is trained within a unit sphere spherical bounding area, the max range is 2, thus $\epsilon_0 = 2$. We can compute the number of the theoretical minimum iteration is 83. However, this limit condition exists only in a very small probability, we set T to 20 through experimental comparison as discussed in Section 5.4.

4 Reflectance Factorization based on L-Tracing

We design a two-stage training framework based on L-Tracing, as shown in Fig. 4, to decompose the material and reflectance of neural surface from posed multi-view images under unknown illumination, enabling novel view interpolation and relighting. Our framework starts from training an neural surface leveraging recent works applying volume rendering to SDF reconstruction. After which we perform reflectance factorization on the well trained shape, with L-Tracing as the light estimation method.

4.1 Shape Learning

Our goal is to learn a neural surface, from which we can extract the 3D mesh with de-illuminated texture and reflectance for rendering. Neural volumetric field shows the strong ability to handle complex geometry occlusions(e.g. the abrupt depth change), however, the free space artifacts prevent neural volume field from modeling well-shaped surfaces. Inspired by the recent works applying volume rendering on the neural implicit surface, we apply SDF as our shape representation, leveraging its simple parameterization, we refer to the [29] for further details. Fig. 4 Stage one shows the process of implicit surface learning.

4.2 Reflectance Factorization

Our Bidirectional Reflectance Distribution Function(BRDF) model is designed similar to NeRFactor [39]. We model the Lambertian reflection fully with the albedo MLP $\boldsymbol{a}$, and the specular spatial-varying component with the BRDF MLP $\boldsymbol{f}_{\mathrm{r}}$. The formulation of our BRDF model is:

$$\boldsymbol{R}\left(\boldsymbol{x}_{\text{surf}}, \boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}}\right) = \frac{\boldsymbol{a}\left(\boldsymbol{x}_{\text{surf}}\right)}{\pi} + \boldsymbol{f}_{\mathrm{r}}\left(\boldsymbol{f}_{\mathrm{z}}(\boldsymbol{x}_{\text{surf}}), \boldsymbol{n}(\boldsymbol{x}_{\text{surf}}), \boldsymbol{\omega}_{\mathrm{i}}, \boldsymbol{\omega}_{\mathrm{o}}\right) \tag{10}$$

The BRDF MLP models the function of the surface point's normal $\boldsymbol{n}(\boldsymbol{x}_{\text{surf}})$, the incoming light direction $\boldsymbol{\omega}_{\mathrm{i}}$, the out-coming light direction $\boldsymbol{\omega}_{\mathrm{o}}$, and the latent material code $\boldsymbol{z}$. According to NeRFactor, the BRDF MLP is pre-trained on the MERL dataset [18] to learn the priors from real-world reflectance. During training, we fix the parameters of BRDF MLP, then recover the surface BRDF by learning the material latent code of the corresponding surface point. We regard the BRDF MLP as the material database and material latent code as the index. By querying the index to the BRDF MLP with related directions and normal, it outputs the final BRDF result.

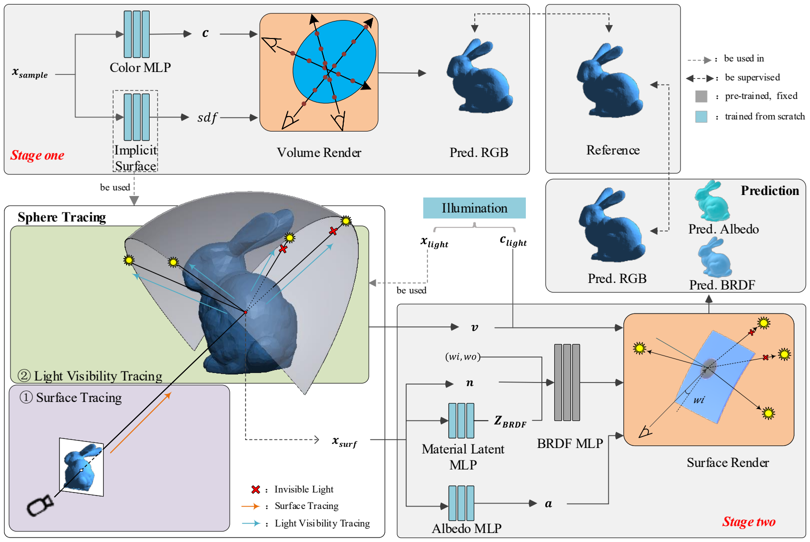


Fig. 4. Overview of Our framework. The training procedure is divided into two stages. **Stage One** is the training of neural implicit surface. The input is the point sets sampled on the rays, $\boldsymbol{c}$ is the space color of each point. The predicted ray color is rendered by volume rendering. **Stage Two** shows how the surface is decomposed. With $\boldsymbol{x}_{\text{surf}}$ obtained by sphere tracing, the light visibility v is computed by L-Tracing, $\boldsymbol{n}$ denotes the surface normal, $\boldsymbol{a}$ is the predicted albedo, BRDF MLP is pre-trained on MERL dataset to learn reflectance priors. The RGB images and Albedo images are rendered by the surface rendering module. In both stages, we use L2 RGB loss to compute the gradients of the networks.

Our illumination is an HDR light probe image [7] in the latitude-longitude format. The light sources are located at a sphere, m denotes their total number. $\boldsymbol{L}_{\text{j}}$ denote the color of the j_{th} light source. Obeying the physical energy law of rendering [11], we define A_{j} as the area of the j_{th} light source, indicating the proportion of its energy on the entire sphere, thus $\sum_0^{m-1} A_{\text{j}} = 4\pi$.

We directly use L-Tracing to estimate the light visibility while training. As shown in Fig. 4, we first compute the intersection surface points $\boldsymbol{x}_{\text{surf}}$ (in the following, we use $\boldsymbol{x}_{\text{s}}$ for short), then apply L-Tracing to estimate the light visibility $V = \{v_0, v_1, \dots v_{m-1}\}$. Since we train our reflectance decomposition under unknown illumination, the predicted illumination greatly contributes to our final output, we only model one-bounce reflection in our work. After we obtain the surface point $\boldsymbol{x}_{\text{s}}$, the corresponding surface normal computed from SDF gradient, the predicted albedo, and the material latent code, we render an image with learned illumination by surface rendering. The render formulation is:

$$\boldsymbol{L}_{\text{o}}\left(\boldsymbol{x}_{\text{s}}, \boldsymbol{\omega}_{\text{o}}\right) = \sum_{\text{j}=0}^{m-1} \boldsymbol{R}\left(\boldsymbol{x}_{\text{s}}, \boldsymbol{\omega}_{\text{i}}, \boldsymbol{\omega}_{\text{o}}\right) \ \left(\boldsymbol{\omega}_{\text{i}} \cdot \boldsymbol{n}(\boldsymbol{x}_{\text{s}})\right) \cdot v_{\text{j}} \cdot \frac{A_{\text{j}}}{4\pi} \cdot \boldsymbol{L}_{\text{j}} \tag{11}$$

This is the integral over the sphere in a discrete form, indicating the color of the out-coming ray is the sum of the in-coming rays reflected by the surface. In this formulation, light visibility is served as the mask for in-coming lights, indicating whether the surface is visible to the lights. We note the way that the light visibility contributes to the final rendered color explicitly respects the solid object surface and binary geometry occlusions, that is beneficial to casting sharper shadows on the surface.

We use multi-view images as supervision, with L2 photometric loss, and smoothness loss to regularize albedo MLP, Material Latent MLP and the illumination. Similar to NeRFactor [39], we randomly initialize the trainable networks and parameters with the uniform distribution.

5 Experiments

In this section, we test our framework based on L-Tracing quantitatively and qualitatively. We 1)compare L-Tracing with volumetric integration on light visibility estimation, 2)compare our factorization results with alternative approaches, 3)show the novel view interpolation results of ours and other works.

5.1 Light Visibility Estimation

We show the light visibility estimation results derived from I. **L-Tracing on Neural Surface(NS+LT):** our proposed framework based on L-Tracing; II. **Volume Integration on Neural Surface(NS+VI):** based on SDF shape representation, convert signed distance to density for volume integration; III. **Volume Integration on Neural Volume(NV+VI):** based on neural volume representation, directly apply volumetric integration on neural volumes.

Visual Quality. In order to show the limitation and robustness of L-Tracing, alongside "mean", the mean visibility across all lights, the light visibility of single point light with no ambient illumination(One-Light-at-A-Time, OLAT) is listed, too. The results in Fig. 5 show, NV+VI struggles with the noise, especially in OLAT2 because of the free space artifacts. NS+VI tends to produce soft shadows since the light visibility is a continuous number ranging from 0 to 1, modeling the direct illumination incorrectly. NS+LT makes highly noise-free results on different illuminations and recovers sharper shadows since it is based on solid surfaces and binary geometry occlusions.

Computational Efficiency. High computational efficiency for light visibility estimation is our heart contribution. In Tab. 1, we recorded the time cost on light visibility computation of L-Tracing and volumetric integration. The test is done on one NVIDIA RTX 3090 GPU, we input a batch of surface points with different batch size. For L-Tracing, we set the tracing iteration to 20(according to Section 5.4, 20 iterations is sufficient for estimation), for volumetric integration, referring to [39], we set the sampling number of coarse points to 64 and 128 for

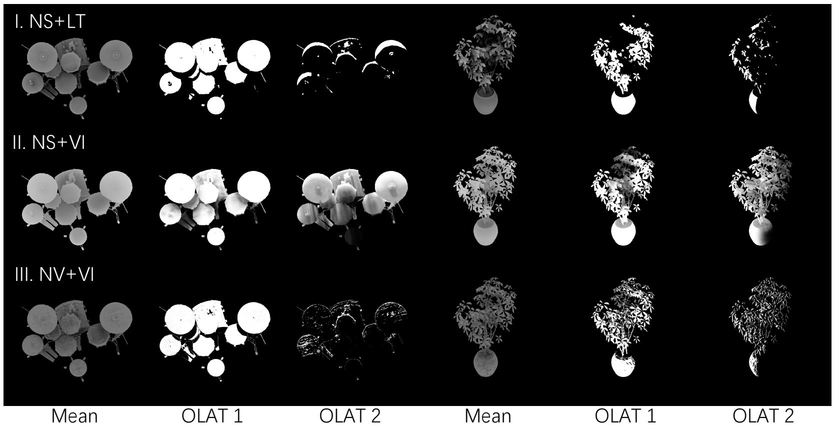


Fig. 5. Light Visibility Visualization. The light visibility is estimated on the NeRFactor synthetic dataset. The color denotes the lighting intensity on the surface. "Mean" denotes the mean light visibility across all light sources, while "OLAT1" and "OLAT2" denotes the light visibility under two different single light source points. Please refer to the supplementary material for more results.

Table 1. Efficiency Comparison. For each surface point, we compute the light visibility of 512 light sources.

Points	L-Tracing	Volumetric Integration
1024	1.07 sec.	13.36 sec.
2048	2.10 sec.	26.86 sec.
4096	4.13 sec.	64.75 sec.

Table 2. Iteration Ablation. Novel view interpolation quality of our framework in different tracing iterations

Iterations	PSNR↑	SSIM↑	LPIPS↓
20	31.194	0.943	0.063
40	31.655	0.944	0.061
80	31.889	0.946	0.060

fine sample. Results show that L-Tracing is about 13 times faster than volumetric integration.

5.2 Reflectance Factorization

We design a framework, as described in section 4, for reflectance factorization. We compare it with Nvdiffrec [22], and NeRFactor [39] under two settings: 1)the original setting **NeRFactor**: precompute light visibility by volume integration, train a light visibility MLP using the precomputed caches. 2)**NeRFactor + VI**: directly use precomputed light visibility caches for factorization.

We visualize the predicted albedos of all methods in Fig. 6. Albedos predicted by L-Tracing based scheme shows similar predictions on details with those from NeRFactor, due to the similar factorization setting. Albedos from Nvdiffrec show uneven color blocks on the object surface, the reason is that the low-level smooth regularization leads the networks to distinguish the surface on the same region into different materials.

We record the metrics of albedo in Tab. 3. As there is an inherent scale ambiguity in inverse rendering problems: the ambiguity between the illuminant's average color and the albedo's average color. To make a fair comparison, we rescaled the albedos of all methods by scaling each RGB channel with a scalar to minimize the mean squared error with ground truth albedos. The result shows that our method performs on par with alternative approaches on albedo recovery across different metrics.

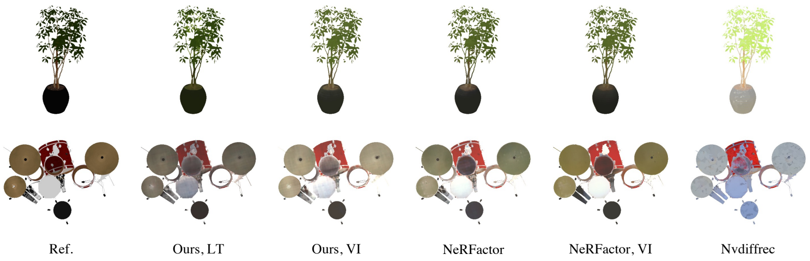


Fig. 6. Albedo Recovery. We show the recovered albedo on NeRFactor synthetic data. The albedos predicted by Ours and NeRFactor are all scaled to minimize the error with ground truth. The rightmost images were provided by Nvdiffrec's [22] authors. We show more recovered albedos in the supplementary material.

5.3 Novel View Interpolation and Relighting

Novel View Interpolation. We record the novel view synthesis quality in Tab. 3, it shows the works specializing in different areas all achieve good results, since the listed method commonly use the multi-view images as supervision, volume rendering based methods show a gift in learning multi-view consistency, which makes the results robust. We note that the L-Tracing based scheme shows little quality reduction on novel view interpolation.

Novel View Relighting. To evaluate the quality of light visibility and recovered material, we relight the learned surfaces and reflectance with different HDR probes. Tab. 3 shows that the L-Tracing based scheme produces comparable relighting quality with recent works. We surpassed all compared methods on SSIM.

We show the relighting results of Lego and Hotdog in Fig. 8. The relighting results are close to the reference intuitively from vision. We select one region on the images of Hotdog under "Sunrise" to compare the relighting details, as shown in Fig. 7. Ours+VI lost the shadows on the left side because the volume integration based light visibility estimation fails to model geometry occlusion correctly. NeRFactor produces smooth shape details and soft shadows because the learned normals and light visibility through MLPs omit the high-frequency

Table 3. Quantitative Evaluation. We calculate the metrics on four scenes of NeRFactor synthetic dataset. Each scene includes 8 novel validation views, for each view, 8 probe relighting images and 1 ground truth albedo are provided. The reported numbers are the arithmetic mean over 4 scenes. Following [22], we rescaled the albedos of all methods before measuring the errors, to eliminate the scale ambiguity. To make a fair comparison, the rendered albedo and relighting images are all blended with the same masks. The results of NeRF are token from the Table 4 of Nvdiffrec [22] paper.

Method	View Interpolation			Albedo			HDR Relighting		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF [21]	31.080	0.956	0.064	—	—	—	—	—	—
NeuS [29]	30.193	0.944	0.081	—	—	—	—	—	—
Nvdiffrec [22]	30.640	0.965	0.044	26.205	0.929	0.078	25.502	0.919	0.073
NeRFactor [39]	31.869	0.944	0.076	27.829	0.943	0.065	26.016	0.915	0.090
NeRFactor,VI [39]	30.288	0.916	0.097	27.686	0.941	0.070	24.913	0.870	0.122
Ours, VI	32.229	0.947	0.059	27.594	0.935	0.076	25.878	0.917	0.075
Ours, LT	31.194	0.943	0.063	27.296	0.933	0.080	25.586	0.920	0.080

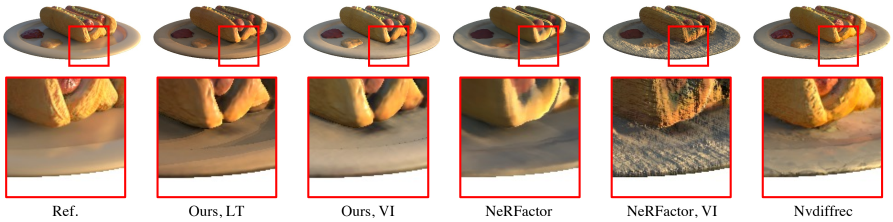


Fig. 7. Relighting Details. We select one area of Hotdog to compare the shape details and relighted shadow. The images are rendered with a new ambient illumination "Sunrise" in the HDR format.

component. NeRFactor+VI directly uses the light visibility integrated on the neural volumes, struggling with the free space artifacts. Nvdiffrec appears to have uneven surface color blocks due to the uneven albedo. L-Tracing based scheme not only learns the detailed surface with high smoothness leveraging the neural implicit surface representation, but also produces sharper shadows, since L-Tracing respects solid object surfaces and binary geometry occlusions.

5.4 Ablation Study

Light visibility $\{0, 1\}$ vs. $[0, 1]$. As L-Tracing estimates light visibility based on geometry occlusions, the binary numbers in $\{0, 1\}$ indicates whether the surface point is visible to the lights. Volume integration assumes the light visibility is the probability the light isn't blocked, thus the number is continuous between 0 and 1. To showcase the effectiveness of these two assumptions. We compare Ours+LT(which is $\{0, 1\}$) with Ours+VI(which is $[0, 1]$). As shown in Tab. 3,

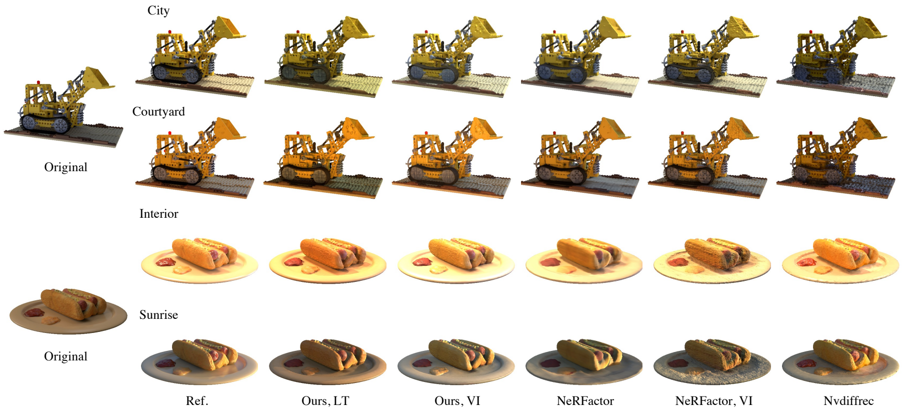


Fig. 8. Novel View Relighting. The factorization results produced by these methods enable us to perform novel view relighting. The illumination representation of listed methods for relighting is HDR probes. We select Hotdog and Lego and relighted with four different HDR probes for comparison, please see supplementary for more.

Ours+LT shows a small drop on albedo and relighting quality, that does not affect the overall visual quality. However, as for light visibility estimation in Fig. 5, NS+VI seems to destroy the physical consistency of observed objects. On drums OLAT2, light sheds on most surface that is invisible to the light source, while NS+LT shows sharp and clear shadows. These comparisons show that L-Tracing out-performs volume integration in terms of the definition of light visibility.

Tracing Iteration. The novel view synthesis results in Tab. 2 show that reducing the iteration from 80 to 20 doesn't significantly reduce the quality of factorization, indicating that it rarely needs 83 iterations(which is the theoretical minimum iterations) to compute the accurate light visibility. We also visualize the light visibility estimated in different tracing iterations in supplementary.

6 Conclusions

We have proposed L-Tracing, a highly efficient but also accurate enough light visibility estimation method on neural implicit surfaces. L-Tracing respects the solid object surface and the binary geometry occlusions, as a result, produces sharper shadows and specular. Moreover, L-Tracing is ~10x faster for light visibility computation than volumetric integration. Furthermore, since our proposed framework models the direct illumination efficiently, we believe that L-Tracing can be further applied to model multi-bounce reflection. All above, it is potential to apply L-Tracing for large scale scene factorization which is highly demanded for computation efficiency and modeling accuracy.

Acknowledgement. The authors Ziyu Chen and Li Song were supported by the Shanghai Key Laboratory of Digital Media Processing and Transmissions.

References

1. Barron, J.T., Malik, J.: Shape, illumination, and reflectance from shading. IEEE transactions on pattern analysis and machine intelligence **37**(8), 1670–1687 (2014) 4
2. Bi, S., Xu, Z., Srinivasan, P., Mildenhall, B., Sunkavalli, K., Hašan, M., Hold-Geoffroy, Y., Kriegman, D., Ramamoorthi, R.: Neural reflectance fields for appearance acquisition. arXiv preprint arXiv:2008.03824 (2020) 1, 4
3. Boss, M., Braun, R., Jampani, V., Barron, J.T., Liu, C., Lensch, H.: Nerd: Neural reflectance decomposition from image collections. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12684–12694 (2021) 1, 2, 4
4. Boss, M., Groh, F., Herholz, S., Lensch, H.P.: Deep dual loss brdf parameter estimation. In: MAM@ EGSR. pp. 41–44 (2018) 4
5. Boss, M., Jampani, V., Braun, R., Liu, C., Barron, J., Lensch, H.: Neural-pil: Neural pre-integrated lighting for reflectance decomposition. Advances in Neural Information Processing Systems **34** (2021) 1, 4
6. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5939–5948 (2019) 3
7. Debevec, P.: Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In: Acm siggraph 2008 classes, pp. 1–10 (2008) 9
8. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised nerf: Fewer views and faster training for free. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12882–12891 (2022) 3
9. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5501–5510 (2022) 3
10. Garbin, S.J., Kowalski, M., Johnson, M., Shotton, J., Valentin, J.: Fastnerf: High-fidelity neural rendering at 200fps. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14346–14355 (2021) 3
11. Kajiya, J.T.: The rendering equation. In: Proceedings of the 13th annual conference on Computer graphics and interactive techniques. pp. 143–150 (1986) 9
12. Knodt, J., Baek, S.H., Heide, F.: Neural ray-tracing: Learning surfaces and reflectance for relighting and view synthesis. arXiv preprint arXiv:2104.13562 (2021) 2, 4
13. Kuang, Z., Olszewski, K., Chai, M., Huang, Z., Achlioptas, P., Tulyakov, S.: Neroic: Neural rendering of objects from online image collections. arXiv preprint arXiv:2201.02533 (2022) 4
14. Lensch, H., Kautz, J., Goesele, M., Heidrich, W., Seidel, H.P.: Image-based reconstruction of spatially varying materials. In: Eurographics Workshop on Rendering Techniques. pp. 103–114. Springer (2001) 4
15. Lensch, H.P., Lang, J., Sá, A.M., Seidel, H.P.: Planned sampling of spatially varying brdfs. In: Computer graphics forum. vol. 22, pp. 473–482. Wiley Online Library (2003) 4
16. Li, Z., Shafiei, M., Ramamoorthi, R., Sunkavalli, K., Chandraker, M.: Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2475–2484 (2020) 4

17. Martin-Brualla, R., Radwan, N., Sajjadi, M.S., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7210–7219 (2021) 3
18. Matusik, W.: A data-driven reflectance model. Ph.D. thesis, Massachusetts Institute of Technology (2003) 8
19. Michalkiewicz, M., Pontes, J.K., Jack, D., Baktashmotlagh, M., Eriksson, A.: Implicit surface representations as layers in neural networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4743–4752 (2019) 3
20. Mildenhall, B., Hedman, P., Martin-Brualla, R., Srinivasan, P.P., Barron, J.T.: Nerf in the dark: High dynamic range view synthesis from noisy raw images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16190–16199 (2022) 3
21. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European conference on computer vision. pp. 405–421. Springer (2020) 3, 13
22. Munkberg, J., Hasselgren, J., Shen, T., Gao, J., Chen, W., Evans, A., Müller, T., Fidler, S.: Extracting triangular 3d models, materials, and lighting from images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8280–8290 (2022) 1, 4, 11, 12, 13
23. Niemeyer, M., Geiger, A.: Giraffe: Representing scenes as compositional generative neural feature fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11453–11464 (2021) 3
24. Oechsle, M., Peng, S., Geiger, A.: Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5589–5599 (2021) 1, 4
25. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 165–174 (2019) 3
26. Srinivasan, P.P., Deng, B., Zhang, X., Tancik, M., Mildenhall, B., Barron, J.T.: Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7495–7504 (2021) 1, 2, 4
27. Sun, C., Sun, M., Chen, H.T.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5459–5469 (2022) 3
28. Verbin, D., Hedman, P., Mildenhall, B., Zickler, T., Barron, J.T., Srinivasan, P.P.: Ref-nerf: Structured view-dependent appearance for neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5491–5500 (2022) 1, 4
29. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. Advances in Neural Information Processing Systems **34**, 27171–27183 (2021) 1, 3, 4, 8, 13
30. Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V.A.: Nerf–: Neural radiance fields without known camera parameters. arXiv preprint arXiv:2102.07064 (2021) 3

31. Wimbauer, F., Wu, S., Rupprecht, C.: De-rendering 3d objects in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18490–18499 (2022) 4
32. Wizadwongsa, S., Phongthawee, P., Yenphraphai, J., Suwajanakorn, S.: Nex: Real-time view synthesis with neural basis expansion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8534–8543 (2021) 1, 2, 4
33. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. Advances in Neural Information Processing Systems **34** (2021) 1, 4
34. Yifan, W., Wu, S., Oztireli, C., Sorkine-Hornung, O.: Iso-points: Optimizing neural implicit surfaces with hybrid representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 374–383 (2021) 3
35. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4578–4587 (2021) 3
36. Zhang, J., Yang, G., Tulsiani, S., Ramanan, D.: Ners: Neural reflectance surfaces for sparse-view 3d reconstruction in the wild. Advances in Neural Information Processing Systems **34** (2021) 1, 4
37. Zhang, K., Luan, F., Wang, Q., Bala, K., Snavely, N.: Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5453–5462 (2021) 2, 4
38. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: Analyzing and improving neural radiance fields. arXiv preprint arXiv:2010.07492 (2020) 3
39. 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 (TOG) **40**(6), 1–18 (2021) 1, 2, 3, 4, 8, 10, 11, 13