

西北工业大学

数字图像处理--论文翻译

原论文标题：从解耦的 NeRF 表示中进行材料变换

李可期

计算机学院

计算机科学与技术

2024 年 11 月

学号：2022302589

从解耦的 NeRF 表示中进行材料变换

试用水印

Ivan Lopes¹ Jean-François Lalonde² Raoul de Charette¹
¹Inria ²Université Laval

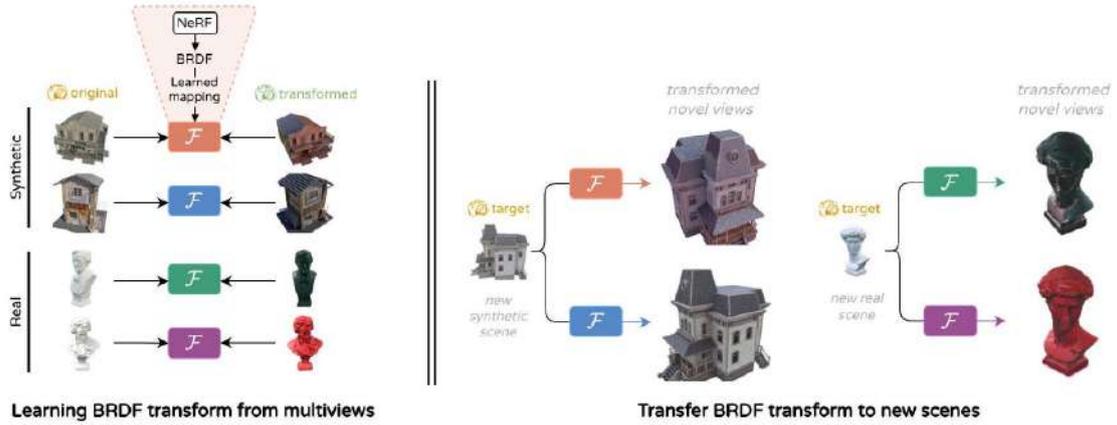


图 1: 提出的方法。我们展示了在复杂场景中推断未知材料转变的方法。从一组场景在两种条件下（原始和转变）的观察中，我们利用联合神经辐射场（NeRF）优化来学习一个材料映射函数 \mathcal{F} ，该函数能够准确地建模材料层次上的观察变化（例如，左侧最上面的转变是红色清漆）。这个学习到的函数可以应用于具有不同几何和材料属性的新目标场景（右侧）。

摘要

在本文中，我们首先提出了一种跨不同场景传递材料变换的新方法。基于解耦的神经辐射场（NeRF）表示，我们的方法学习从在不同条件下观察到的场景对中映射双向反射分布函数（BRDF）。然后可以将学习到的变换应用于具有相似材料的未见场景，从而有效地以任意强度渲染学习到的变换。在合成场景和真实物体上的大量实验验证了我们方法的有效性，显示它可以学习各种变换，如湿度、绘画、涂层等。我们的结果不仅突出了我们方法的多样性，还展示了其在计算机图形学中的实际应用潜力。我们在<https://github.com/astra-vision/BRDFTransform>上发布了我们的方法实现以及我们的合成/真实数据集。

1. 引言

在计算机图形学和视觉中，逆向渲染是提取材料信息并允许在新条件下重新渲染（视点、光照、材料等）的关键。虽然神经表示在很大程度上取代了传统的基于物理的渲染（PBR）技术，但最近的工作表明，这两种表示可以结合使用 [JLX*23]，从而保留 PBR 表示的可编辑性和表现力以及神经表示的灵活性。

当考虑场景的外观时，某些变换（如涂上一层清漆）可以显著改变材料特性，导致场景外观发生巨大变化。目前，估计已知材料在此类变换后的 PBR 特性需要在所需目标条件下重新捕获场景。由于可能的变换种类繁多，如湿度、灰尘、清漆、绘画等，这一过程既复杂又费力。在这项工作中，我们旨在从源场景中学习 BRDF 变换并将其应用于不同的场景。

假设我们有同一场景在两种不同条件下的配对观察，例如原始和清漆，我们提出了一种学习材料变换的方法。然后将这种变换应用于由相似材料组成的另一个场景。这使我们能够预测该场景在这种效果下的外观，从而有效地传递材料变换。

图 1 说明了可以从多个场景对（左侧）中学习到的几种材料变换，并随后应用于新场景（右侧），无论是合成的还是现实的。从技术上讲，我们的方法依赖于对应于在原始和变换（例如，清漆）条件下捕获的第一个场景的辐射场的联合优化，可能不同的光照条件下。我们在这里依赖于 TensorIR 的解耦 NeRF 表示 [JLX*23]，它同时优化外观、几何和参数化 BRDF，同时引入两个新的关键组件。首先，我们将变换后的场景 BRDF 条件化在原始场景上，并用多层感知机（MLP）近似其变换。其次，我们揭示了 TensorIR 的一个局限性，显示它在分解高反射材料时失败，并提出了一种改进的光估计方案，该方案更好地估计低粗糙度成分，同时保留光照中的高频。结果是，我们的框架允许捕获一系列变换，然后可以应用于新场景，同时控制变换的强度。我们在两个新数据集上展示了我们方法的性能：一个具有一系列自定义着色器变换的合成数据集和一个具有不同材料条件（例如，原始、绘画、清漆等）的真实世界数据集。在这两个数据集上，我们的方法产生了真实的变换。我们的方法和数据集将公开发布。

2. 相关工作

逆向渲染是一个长期存在的问题，最近随着神经辐射场的使用而引起了兴趣 [MST*20]。给定从不同视点拍摄的物体的一组图像，目标是优化不透明度和辐射的隐式体积模型。这允许使用体积渲染在新视点下合成帧。许多工作扩展了这种方法，以学习一个更明确的体积，其中材料信息与光源解耦。这样，场景可以重新点亮并操纵材料，提供比传统的以辐射为中心的方法更多的控制。

NeRF 中的 BRDF 估计。NeRD [BBJ *21] 是第一个在不受控环境中执行场景 BRDF 优化的方法。后来, 诸如 NeRV [SDZ *21] 和 IndiSG [ZSH *22] 的方法引入了解决自遮挡和间接光的解决方案。球面高斯 (SG) 已被广泛用于逆向渲染中的光照建模 [ZLW* 21, ZSH*22, JLX*23, ZXY* 23]。随后在 NeILF/++ [YZL22, ZYL23]、NeRO [LWL23] 和 TensoSDF [LWZW24] 中引入了隐式表示, 以更好地表示高频光照。对于镜面物体, 有些人提出了新的编码形式来帮助监督狭窄的镜面叶片。例如, RefNeRF [VHM22] 使用集成方向编码 (IDE), NeAI [ZZW24] 使用集成叶片编码 (ILE), SpecNeRF [MAT24] 使用高斯方向编码。这些优化方法已与 Factored-NeuS [FSV23] 和 NeRO [LWL23] 中的符号距离函数 (SDF) 结合使用, 以提供更稳健的几何估计。最近, NeP [WHZL24] 使用神经全光函数来建模入射光。与采用分析 BRDF 的其他方法不同, NeRFactor [ZSD* 21] 通过首先从 MERL 数据集 [MPBM03] 学习真实世界 BRDF 的先验来使用数据驱动的方法。相反, ENVIDR [LCL23] 在合成数据集上学习这种先验。NVDiffrec/-MC [MHS22, HHM22] 优化网格及其材料作为 SVBRDF 映射。

在场景优化方面最接近我们的方法, TensorIR [JLX* 23] 采用张量表示并因子化光组件以在多种光照下学习。他们使用分层采样和 SGs 来建模直接光, 而我们采用神经表示。NeRO [LWL*23] 采用相同的光照方法, 但使用计算成本高的两阶段方法。相反, 我们使用渲染方程的近似来预计算积分的一部分。

材料和神经变换。虽然在多视图设置中 BRDF 变换的问题尚未被探索, 据我们所知, 我们在这个主题上提出了相关研究。在 tvBRDF [SSR07] 中, 作者提出了分析模型, 用于在非空间变化材料上进行变换, 如灰尘、水彩、油和喷雾。另一条研究线着眼于基于示例风格图像翻译 NeRF 重建。这包括 StyleNeRF [LZC 23]、LAENeRF [RSKS24] 或 iNeRF2NeRF [HTE23], 这是基于提示的。还相关的是执行材料转移的任务, 如在 NeRFanalogies [FLNP 24] 中。在 Climate-NeRF [LLF* 23] 中, 全局效果被注入到场景中, 但不影响物体材料。

逆向渲染数据集。具有不同 BRDF 的数据集在 [GTR06] 中引入, 具有时间变化效果。他们记录了一些由自然过程转变的表面, 展示了它如何在时间和空间上影响 BRDF。通常, 逆向渲染数据集提供在不同光照下的捕获, 但材料保持不变: ReNe [TMS*23] 提出了一组在 40 个点光源位置下捕获的 20 个真实场景的数据集。Objects-with-Lighting [UAS24] 引入了 8 个物体在 3 个环境下的捕获, 并提供了相应的高动态范围 (HDR) 环境图。

3. 方法

3.1. 问题设置

考虑图 2 中显示的场景，其中一个场景在其原始状态 s_0 下观察，第二次 s_1 其材料被未知效果 T 转变，使得 $s_1 = T(s_0)$ 。例如， T 可能是涂上一层油漆、一些彩色清漆或场景被水浸泡的结果。注意， s_1 可能在与 s_0 不同的光照下被捕获。我们的目标是建模 s_0 和 s_1 之间发生的材料变换，以便我们可以将这种效果转移到新场景中。

我们用 BRDF 场建模场景，更具体地说，场景 s 的每个点都由材料属性 $\beta = (\rho, r) \in \mathbb{R}^4$ 表征，其中 ρ 是反照率 (RGB) 和 r 是粗糙度。我们的公式假设原始场景 s_0 受到

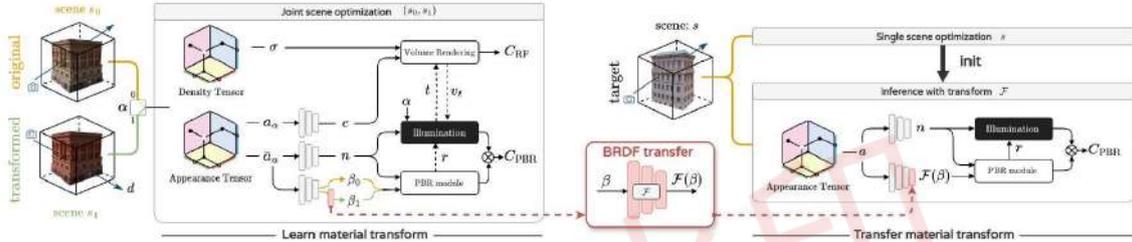


图 2: 我们提出的方法概述。我们的方法获取同一场景在两种不同材料 (β_0, β_1) 下的观察，分别为 s_0 和 s_1 。我们假设 β_1 是 β_0 的函数。我们的方法学习一个联合表示和一个变换函数 \mathcal{F} ，将第一个材料映射到第二个 (左块)。给定一个新场景 s ，我们学习其几何和材料并应用我们学习的变换函数 (右块) 以产生在源场景 (s_0, s_1) 中观察到的相同效果。

受未知变换影响，改变其材料属性 β_0 但不改变其几何，导致场景 s_1 具有材料 β_1 。我们的方法旨在学习一个函数 \mathcal{F} ，近似两个材料 β_0 和 β_1 之间的未知映射 T ，以便 \mathcal{F} 可以应用于新场景，如图 2 (右) 所示。

3.2. 预备知识

我们的优化方法基于 TensorIR [JLX23]，本身源自 TensorRF [CXG22]，以学习场景的神经辐射场。为清晰起见，我们在此遵循他们的符号。在此框架中，通过联合训练密度张量 \mathcal{G}_σ 和外观张量 \mathcal{G}_a 来学习辐射场。从后者中，可以使用轻量级 MLP 估计每个 3D 点 x 的表面法线 n 和材料属性 β ，并沿每个视线使用体积渲染累积。虽然场景可以在单一光照条件下成像，但 TensorIR 也支持在不同光照下对场景的多次观察。在这种情况下，它进一步因子化光嵌入以产生光依赖的外观特征 a_α ，其中 α 索引光照条件 (建模为环境图)。因此，场景中每个点 x 的估计量可以写为：

$$n = \mathcal{D}_n(\bar{a}_\alpha), \quad \beta = \mathcal{D}_\beta(\bar{a}_\alpha), \quad c_\alpha = \mathcal{D}_c(a_\alpha) \quad (1)$$

其中 \bar{a}_α 是跨两个光嵌入的平均外观特征, c 是像素颜色 (如原始 TensoRF 公式中)。TensoIR 学习解耦表示, 允许每个点 x 的颜色通过体积渲染估计, 表示为 $C_{\text{RF}}(x, d)$, 或通过基于物理的渲染, 也表示为 $C_{\text{PBR}}(x, d)$ - 两者都由参考图像监督。

3.3. 学习材料变换

如第 3.1 节所述, 我们旨在学习 \mathcal{F} , 将场景 s_0 的 BRDF 参数 β_0 映射到其变换后的外观 β_1 , 以用于 s_1 。如图 2 所示, 我们将转移公式化为:

$$\beta_\alpha = \beta_0[\alpha = 0] + \mathcal{F}(\beta_0)[\alpha = 1] \quad (2)$$

其中 $[\bullet]$ 是 Iverson 括号, \mathcal{F} 是一个小型 MLP 网络, 与外观和密度张量一起端到端训练。这里 α 是一个指示符, 表示我们是在渲染原始场景 (即 $\alpha = 0$) 还是其变换版本 (即 $\alpha = 1$)。使用这种公式, 我们在 s_0 和 s_1 上联合训练, 并为两个场景学习一个神经表示。

3.4. 光估计

TensoIR 的局限性。我们观察到原始 TensoIR 框架在重建低粗糙度场景 (图 3) 时存在困难, 这对于表示光滑表面至关重要。我们还注意到, 用于表示环境的球面高斯数量较少, 导致光照中缺乏高频内容。使用分层采样和低频光表示的代价是对低粗糙度物体的错误估计。

为了解决这个问题并允许学习更广泛的材料变换, 我们提出了一种改进的公式, 借鉴了 NeRO [LWL*23] 的想法。我们保留 TensoIR 的体积表示, 因为它优化速度快, 但通过遵循 NeRO 避免昂贵的光采样。这样, 我们受益于这两种方法, 并确保快速优化速度。

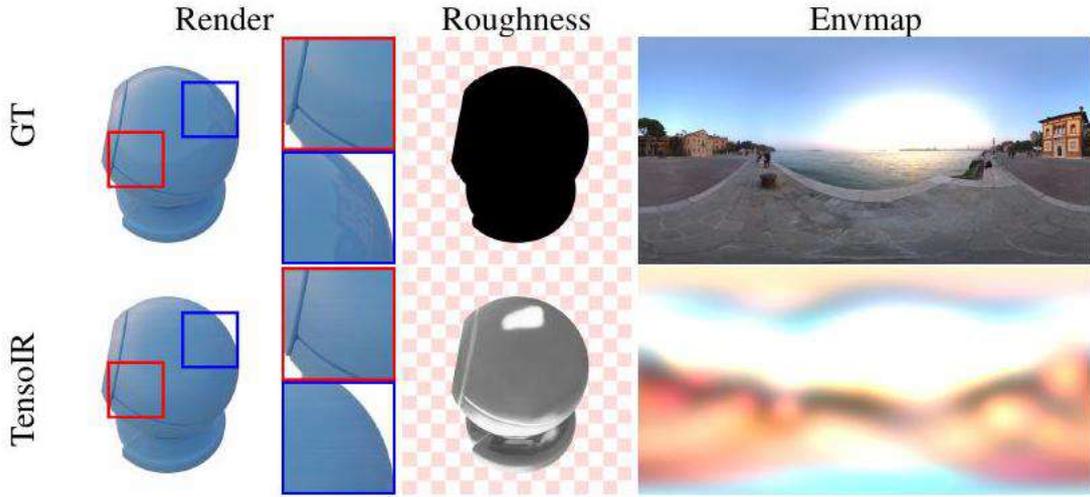


图 3: TensorIR 在光滑表面上。我们观察到 TensorIR 高估了粗糙度并平滑了估计的光照。

公式。从视线方向 d 渲染点 x 的颜色由

$$C_{\text{PBR}}(x, d) = \int_{\Omega} L(\omega, x) f_r(\omega, d) (\omega \cdot n) d\omega, \quad (3)$$

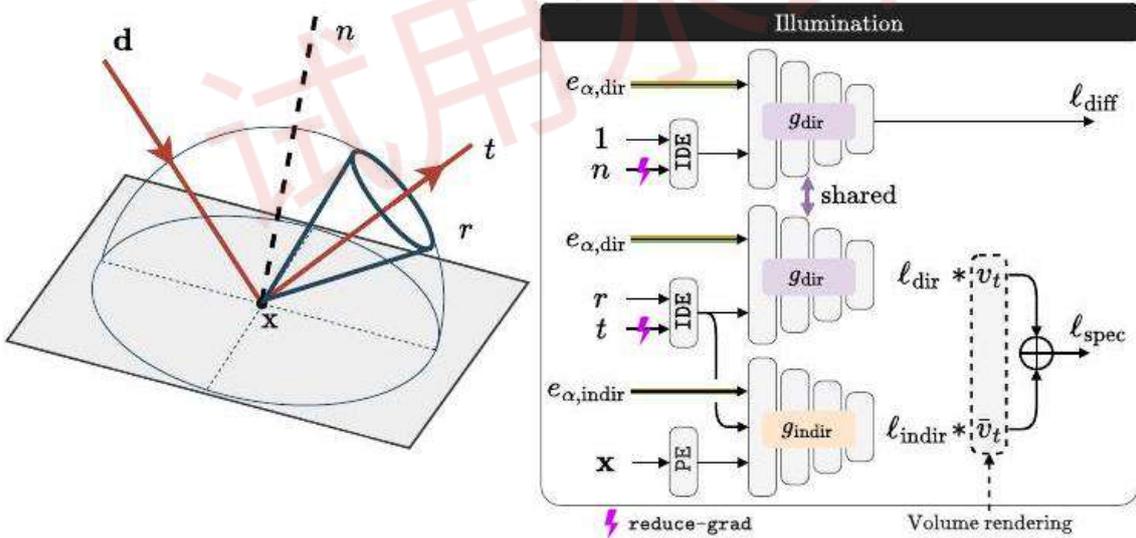


图 4: 光估计。我们采用神经光表示 [LWL* 23], 分别建模直接和间接光源。在间接组件上, 使用通过沿反射光方向进行二次光线投射获得的遮挡掩码混合两种类型的光源 t [JX*23]。为了避免扰乱几何的优化, 我们减少了方向输入 (在 n 和 t 上) 的梯度强度。我们注意到 $\bar{v}_t = 1 - v_t$, IDE 是集成方向编码 [VHM* 22], 而 PE 是位置编码 [MST*20]。

其中 Ω 是积分半球, L 是从方向 ω 在 x 处的光强度。这里, BRDF f_r 由材料属性 $\beta = (\rho, r)$ 参数化。我们采用 [CT82] 的微面反射模型:

$$f_r(\omega, d) = \frac{\rho}{\pi} + \frac{DFG}{4(\omega \cdot n)(d \cdot n)}, \quad (4)$$

其中 D, F 和 G 是法线分布、菲涅耳和几何衰减项。为简洁起见，我们省略了这三个函数的参数。我们遵循 NeRO [LWL*23] 并在镜面组件上使用分裂和求和近似 [Kar13]。积分后，它变为：

$$C_{\text{PBR}}(x, d) = \rho \ell_{\text{diff}} + M_{\text{spec}} \ell_{\text{spec}} \quad (5)$$

其中

$$\ell_{\text{diff}} = \int_{\Omega} L(\omega, x) D(n, 1) d\omega, \quad \ell_{\text{spec}} = \int_{\Omega} L(\omega, x) D(t, r) d\omega \quad (6)$$

和

$$M_{\text{spec}} = \int_{\Omega} \frac{DFG}{4(d \cdot n)} d\omega \quad (7)$$

这里， t 是相对于表面法线 n 的反射方向。注意， M_{spec} 可以预先计算，因为它不依赖于 L 。积分 ℓ_{diff} 和 ℓ_{spec} （依赖于 L ）将在接下来讨论。

光估计。我们使用 Ref-NeRF [VHM22] 的集成方向编码 (IDE) 来建模场景光照。类似于 NeRO [LWL23]，我们利用两个 MLP 来近似 L ，分别为 g_{dir} 用于直接光和 g_{indir} 用于间接（例如，间接反射）光。为了适应两个场景的联合优化设置，我们将潜在嵌入馈送到两个光 MLP g 中，以考虑光照可能的变化。这是通过通道级连接相应的嵌入到 g 输入的 IDE 来实现的，具体取决于渲染原始场景还是变换后的场景。光照表达式写为

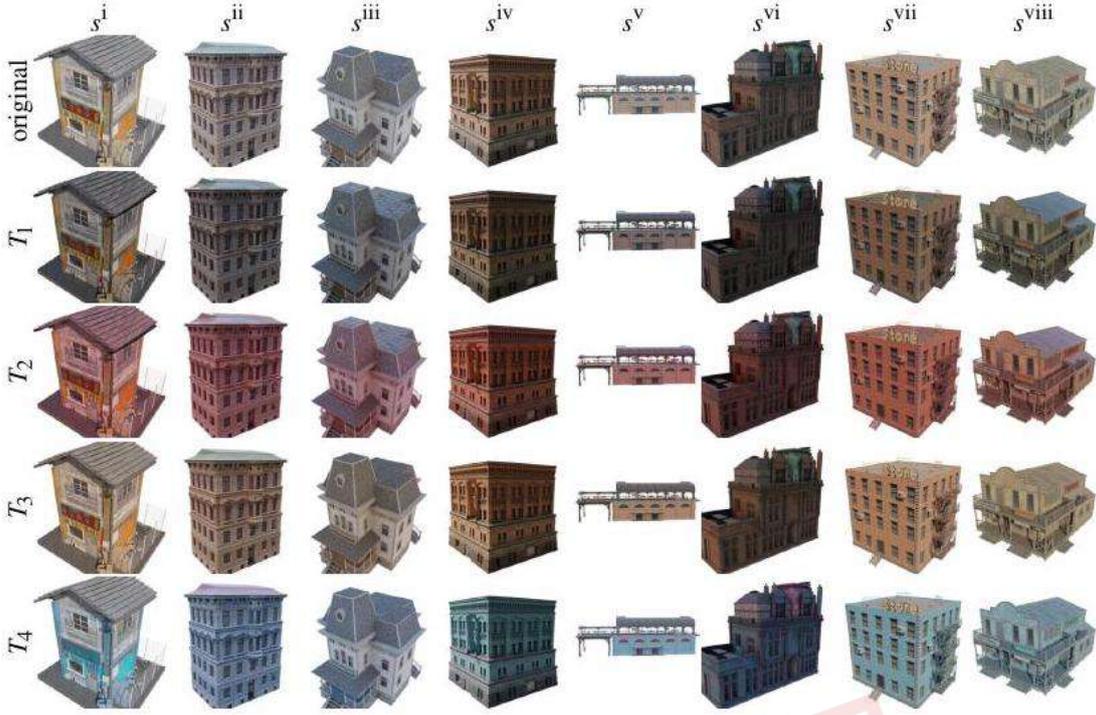


图 5: 合成数据集。每列显示一个不同的场景 $s^k, k \in \{i, \dots, viii\}$ 。第一行显示原始场景，每个后续行显示应用每个合成变换 $T_j, j \in \{1, \dots, 4\}$ 后的场景。



图 6: 真实世界数据集。不同的半身像雕像首先拍摄有和没有各种彩色涂层 (贝多芬、大卫、舒伯特、肖邦、瓦格纳、巴赫) 或光泽清漆 (莫扎特、缪斯)。

$$\begin{aligned}
 \ell_{\text{diff}} &= g_{\text{dir}}(\text{IDE}(n, 1), e_{\alpha, \text{dir}}) \\
 \ell_{\text{spec}} &= v_t g_{\text{dir}}(\text{IDE}(t, r), e_{\alpha, \text{dir}}) \\
 &\quad + (1 - v_t) g_{\text{indir}}(\text{IDE}(t, r), x, e_{\alpha, \text{indir}})
 \end{aligned} \tag{8}$$

对于镜面项 ℓ_{spec} , v_t 是通过沿 t 在密度体积中进行光线追踪获得的可见性项。

图 4 中显示了所提出的光照组件的详细信息。这里，潜在嵌入 e 为 g 网络提供条件，以便最好地建模场景特定的光照。

4. 实验

4.1. 实验方法

鉴于缺乏适合研究 BRDF 转移的数据集，我们构建了两个数据集：一个合成的，具有自定义 Blender 着色器；一个真实的，捕捉在不同材料条件下的雕像。两个数据集都将公开共享。鉴于获取真实世界的地面真实分解的复杂性，我们遵循通常的做法 [JLX*23]，仅在合成数据集上报告定量性能。定性结果显示在两者上。

合成数据集。我们获得了八个免费和开源的 3D 模型，兼容 Blender PBR 渲染管道。模型单独处理并重新缩放，以便它们共享相似的大小。我们在 Blender 中设计了一组简单的着色器变换 $T \in \{T_1, \dots, T_n\}$ ，以改变给定场景中的所有材料。这使我们能够控制场景的全局 α (0: 无变化, 1: 完全变换) 和应用的变换类型。遵循 NeRFactor [ZSD*21] 的数据集创建程序，我们为每个场景渲染 100/20 个训练/测试视图。数据集的概述在图 5 中提供，显示了所有场景和合成变换。接下来，我们简要描述不同的合成变换：

- 在“原始”中，使用原始 PBR 材料；
- T_1 : $r' = 0$ 且 ρ' 具有 ρ 的 30% 的 HSV 值；
- T_2 : $r' = 0$ 且 $\rho' = 0.5\rho + 0.5\rho_{\text{red}}$ ；
- T_3 : $r' = 1$ 且 $\rho' = 0.2\rho + 0.8\rho_{\text{sand}}$ ；
- T_4 : $r' = r$ 且 ρ' 具有 ρ 的相反色调。

其中 ρ_{sand} 和 ρ_{red} 是任意选择的两种 RGB 颜色。虽然现实的变换需要复杂的着色器，但我们强调我们的变换选择是基于对现实世界变换的视觉近似，分别是：湿度 (T_1)、新鲜绘画 (T_2)、尘土 (T_3)、绘画 (T_4)。

真实世界数据集。我们收集了八个半身像雕像（大约 10 厘米高，见图 6）并使用手机拍摄了它们的全貌。我们在两种不同的条件下捕捉它们的外观：首先是它们的原始外观，然后是在改变其材料条件后，例如涂上各种彩色涂层或清漆。与合成数据不同，捕捉真实世界数据会导致未知的相机姿态和两组未对齐的照片，因为它们无法从完全相同的视点拍摄。这可以通过使用专门设计的相机设备来防止，如 [TMS*23] 中所示，但需要花费和时间来构建和校准设备。相反，如图 7 所示，我们分别对原始和变换后的图像集应用 COLMAP [SF16, SZPF16]，然后使用迭代

最近点 (ICP) 算法估计两个结果点云之间的刚性变换矩阵。将结果矩阵应用于将两组相机姿态校正到唯一的参考基准。我们的优化要求我们遮罩场景背景，为此我们使用库 rembg [Gat20] 并手动校正未正确遮罩的帧。

训练。我们在混合批次中训练，使用来自 s_0 和 s_1 的像素光线。所有组件都端到端优化。目标场景的优化是在没有光照嵌入 e_α 的情况下进行的。要在新场景上应用学习到的变换函数，我们只需插入我们训练的 MLP 并计算 $\mathcal{F}(\beta)$ ，如图 2 右侧所示。

网络。为了实现我们在第 3 节中描述的方法，我们采用与 TensoIR [JLX*23] 相同的基础架构和优化程序。向量 e_α 是大小为 72 的光嵌入，用于编码特定于每个 α 的场景信息，因为两个观察 s_0 和 s_1 可能在略有不同的光照条件下捕获。与 TensoIR 一样，我们使用二次光线行进来估计可见性掩码 v_t ，但不是采样辐射场，而是利用专用的 g_{indir} MLP 来估计遮挡方向的辐照度。在反向传播过程中，reduce-grad 函数在梯度上应用 10^{-2} 的权重，以减少其对方向输入的影响。用于学习变换函数 \mathcal{F} 的模型是一个小型 MLP，具有一个隐藏层，维度为 256。

基线。据我们所知，没有基于图像的材料变换学习方法。因此，我们尽最大努力从现有技术中构建强大的基线。

首先，我们注意到 TensoIR [JLX*23] 可以通过将 \mathcal{D}_β 的输入从 \bar{a}_α 替换为 a_α 来适应特定于材料的信息。然而，这带来了两个问题。首先， a_α 不可解释，因为它包含与几何、材料和光照相关的纠缠信息；其次，将此功能转移到新场景将失败，因为两个场景的外观特征将属于不同的嵌入空间。

因此，我们选择了不同的方法来设置一个公平的基线。我们首先分别在原始场景 s_0 和变换场景 s_1 上进行训练。然后，我们通过查询体积提取两个场景的几何和 BRDF。训练一个 MLP 模型来学习两组 BRDF 之间的映射。最后，将此模型应用于新场景 s ，以便将其材料从 β 映射到 $\mathcal{F}(\beta)$ 。我们对几种方法遵循此步骤：在原始 TensoIR 本身 [JLX23] 以及在两个最近的逆向渲染方法：NeRO [LWL23] 和可重光 3D 高斯 (RG3D) [GGL*23] 上。由于某些基线预测了额外的金属成分，而我们没有，因此在优化和新视图合成期间将其值设置为零，以避免不公平的优势。

4.2. 主要结果

学习 BRDF 转移。在表 1 中，我们展示了每个变换 T 的 BRDF 转移结果，平均在所有合成场景上，并在所有变换上取平均。为了评估我们的分解和材料转移的质量，我们报告法线的平均角误差 (MAE) (\downarrow) 和 PSNR(\uparrow), SSIM(\uparrow), 以及

Transfer	Method	$\frac{\text{Normals}^{**}}{\text{MAE}_{\downarrow}}$	Albedo			Render
			PSNR $_{\uparrow}$	SSIM $_{\uparrow}$	LPIPS $_{\downarrow}$	PSNR $_{\uparrow}$
	NeRO	7.726	16.03	0.676	0.254	20.40
	TensoIR	10.92	17.82	<u>0.722</u>	0.262	21.04
T_1	R3DG	11.41	11.57	0.610	0.223	26.12
	ours	6.750	19.80	0.781	0.195	21.95
	NeRO	7.726	16.54	<u>0.693</u>	0.242	21.47
T	TensoIR	10.92	16.66	0.688	0.251	20.02
T	R3DG	11.41	12.44	0.635	0.219	27.36
	ours	6.750	18.62	0.770	0.199	22.44
	NeRO	7.726	16.21	0.690	0.248	24.00
T	TensoIR	10.92	16.90	0.697	0.293	23.72
T	R3DG	11.41	12.79	0.656	0.217	31.45
	ours	6.750	19.81	0.787	0.197	29.76
	NeRO	7.726	15.43	0.662	0.263	22.60
	TensoIR	10.92	17.52	<u>0.696</u>	0.273	22.61
T_4	R3DG	11.41	13.26	0.668	0.217	29.12
	ours	6.750	18.88	0.766	0.203	26.88
	NeRO	7.726	16.05	0.680	0.252	22.12
mean	TensoIR	10.92	17.23	<u>0.701</u>	0.270	21.85
mean	R3DG	11.41	12.52	0.642	0.219	28.51
	ours	6.750	19.28	0.776	0.199	25.26

- 法线与学习的变换无关。

表 1: 新视图转移评估。我们通过跨场景转移后测量指标来评估我们合成数据集上的材料估计。在测试集上对新视图合成进行评估。我们突出显示最佳和第二最佳。

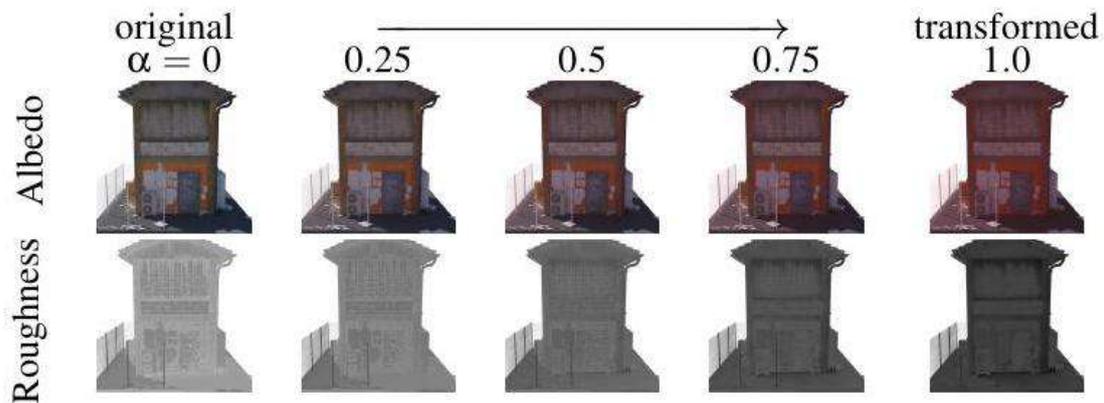


图 8: 变换插值。我们在原始场景和目标场景之间线性插值 BRDF 参数, 针对不同的 $\alpha \in [0, 1]$ 值。

LPIPS (\downarrow) 用于估计的反照率。为了完整性, 我们还报告了新视图合成的渲染 PSNR(\uparrow)。表中的结果表明, 我们的方法更好地估计法线和反照率, 展示了它忠实地学习了 BRDF 函数的变换。重要的是, 注意到 R3DG 具有更高保真度的渲染, 尽管这以不准确的分解 (法线、反照率) 为代价, 这表明场景光照中存在纠缠的材料信息。相应的定性结果显示在图 14 和图 15 中, 展示了四个场景和三种变换。后者展示了我们方法的优越性, 产生了真实的渲染, 而不影响场景分解 (法线、反照率、粗糙度)。在图 8 中, 我们进一步展示了我们的公式允许在原始 ($\alpha = 0$) 和学习的变换 BRDF ($\alpha = 1$) 之间进行插值, 根据公式 (2)。

此外, 我们提供了每个场景反照率性能的热图, 如图 9 所示, 提供了更细粒度的分析。总体而言, 热图表明某些场景更容易

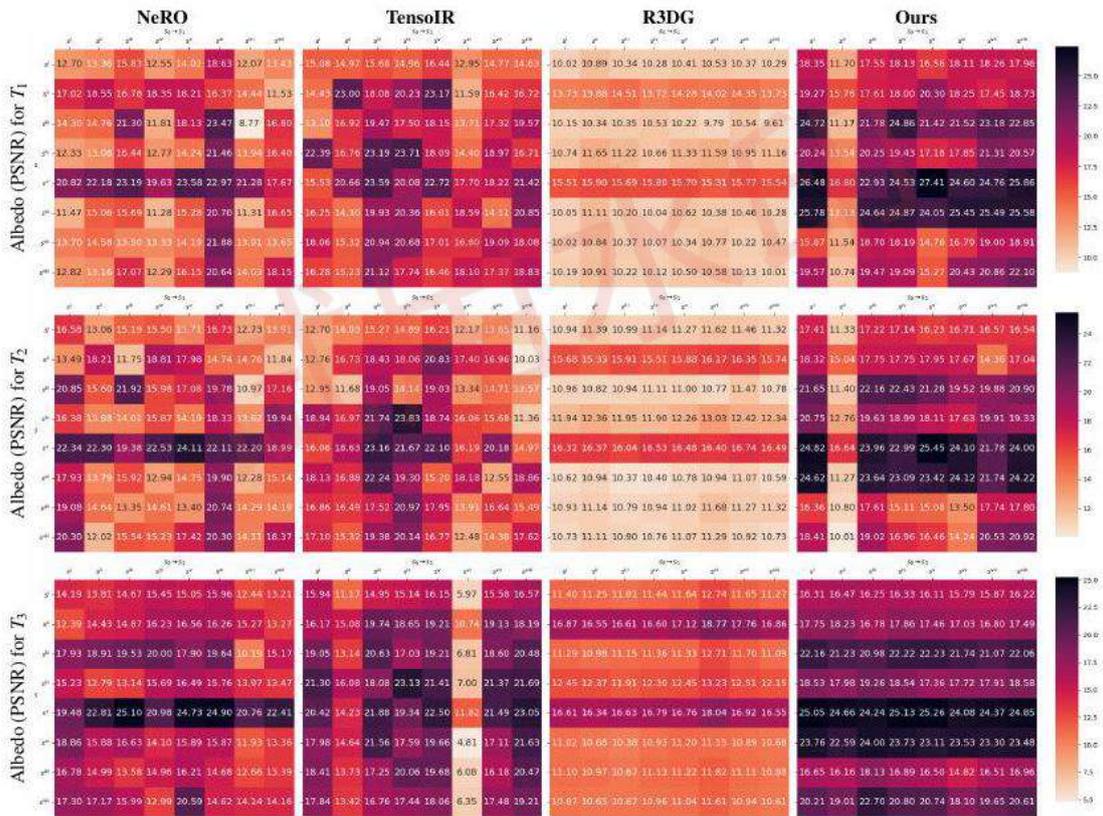


图 9: 每种变换的 BRDF 转移性能。我们提供反照率 PSNR (\uparrow) 的热图, 用于源 (水平) 和目标 (垂直) 场景对。对角线表示在同一场景上应用 BRDF 变换时的性能。值得注意的是, 某些场景很容易转变 (s^{ii}, s^{ii}), 可以说是因为外观更简单 (参见图 5), 而 TensoIR 似乎在学 s^{vi} 场景上的 T_3 时遇到了困难。尽管具有出色的渲染能力 (参见表 1), R3DG 在忠实分解场景方面遇到了困难, 而我们的方法在几乎所有实例中都优于所有基线。

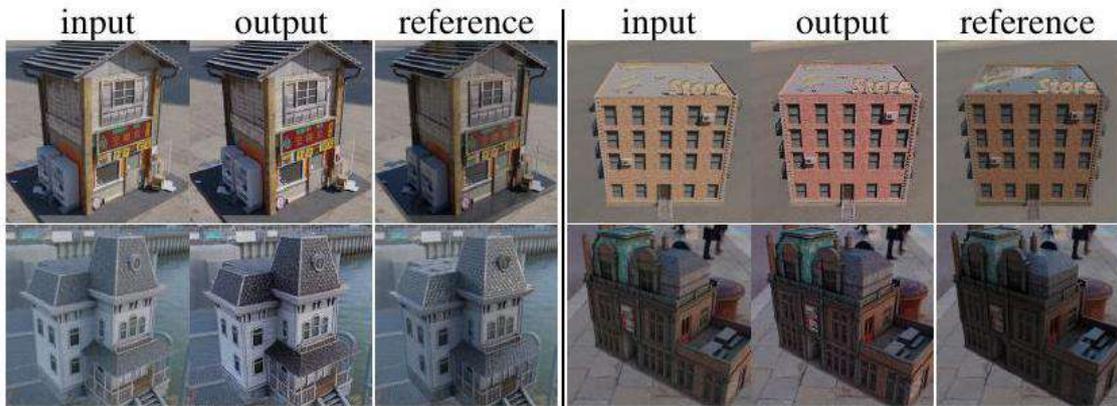


图 10: 基于提示的翻译示例。在带有背景的图像上应用 InstructPix2Pix [BHE23], 输入提示为“使其更光滑”。图像翻译方法限制了输出在几何和外观方面的一致性。它还假设变换是已知的, 并且可以用自然语言表述, 这并不总是如此。变换, 例如 s^{ii} 和 s^{vi} (参见图 5 以供参考), 而从 s^{vi} 学习被证明更复杂 - 特别是对于 TensoIR 和变换 T_3 。如表 1 所示, R3DG 在正确分解反照率方面遇到了困难, 而我们的方法在几乎所有实例中都优于所有基线。

与 i2i 翻译方法的比较。此外, 我们在图 10 中提供了一些使用提示引导的图像到图像 (i2i) 翻译方法 InstructPix2Pix [BHE23] 的结果。结果展示了此类技术的局限性, 这些技术需要关于变换及其自然语言表述的先验知识。此外, 输出在几何上不一致。

4.3. 消融研究

我们在第 4.2 节的合成数据集上进行消融研究, 因为有可用的地面真实材料属性。与之前一样, 我们报告估计的反照率和新视图合成的指标。在这里, 我们将我们提出的方法命名为“完整模型”; “w/o reduce-grad” 允许梯度在光照模块的方向输入上流动而不进行阻尼; “w/o g_{dir}, g_{indir} ” 删除光照 MLP 并使用球面高斯和分层采样来表示光源 (如 TensoIR 中); “w/o joint optim.” 对应于分别学习两个场景表示并拟合 MLP 以学习 \mathcal{F} ; “w/o transfer” 作为下限, 计算原始场景 s_0 的指标与 s_1 的参考图像进行比较。

设计选择对转移能力的影响。在表 2 中, 我们展示了组件对将学习到的变换转移到其他场景的能力的评估。使用神经光表示改善了几何估计以及预测反照率的整体质量。我们注意到, 在原始和变换场景上学习有利于 \mathcal{F} 的转移。

Ablations	Normals	Albedo			Render
	MAE \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow
ours (full model)	6.750	19.80	0.781	0.195	21.95
\hookrightarrow w/o $g_{\text{dir}}, g_{\text{indir}}$	<u>9.060</u>	18.51	0.784	0.187	21.04
\sqcup w/o joint optim.	11.14	17.75	0.714	0.264	20.50

表 2: 转移到其他场景。评估变换 T_1 , 在将 \mathcal{F} 应用于新场景 s_0 后进行新视图合成。结果场景与使用地面真实变换 T_1 渲染的 s_1 的参考图像进行比较。

联合训练的好处。研究在将学习到的 BRDF 变换应用于同一场景（而不是之前所做的新场景）时每个组件的影响也很有趣。这允许评估在源场景上学习的 BRDF 变换的质量。此实验的结果显示在表 3 中。这表明在场景优化期间学习变换函数的好处。我们注意到“w/o transfer”上的 MAE 略好，因为几何是在原始场景上学习的，这通常比变换场景更容易估计。

Ablations	Normals	Albedo			Render
	MAE \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow
ours (full model)	7.164	21.40	<u>0.805</u>	<u>0.187</u>	30.51
\longrightarrow w/o $g_{\text{dir}}, g_{\text{indir}}$	9.338	23.80	0.851	0.210	29.58
\hookrightarrow w/o joint optim.	11.14	19.43	0.747	0.242	22.37
\hookrightarrow w/o transfer	6.750	18.56	0.769	0.175	21.12

表 3: 同一场景的变换。我们测量联合优化 s_0 和 s_1 的增益，并消融不同的组件。在“w/o joint optim.”中，两个场景分别优化，而最后一个实验“w/o transfer”是通过直接从 s_0 评估而不进行转移。

reduce-grad 的效果。当采用集成方向编码 (IDE) 用于光照组件时，我们注意到法线趋于退化。考虑到较短的梯度路径，光 MLP 更容易在以牺牲表面法线和反照率为代价的情况下烘焙反照率信息。我们可以在图 11 的“w/o reduce-grad”列中看到这一点：广告牌（参见红色放大区域）在光中烘焙了高频细节，而它应该是一个完全平坦的表面。通过提出的 reduce-grad 操作符阻尼梯度，防止法线过拟合到光梯度信号，并导致更均匀的漫反射光估计 l_{diff} 。

与 TensoIR 的比较。TensoIR [JLX*23] 中使用的光照围绕分层采样，这不允许渲染低粗糙度表面。因此，它无法建模反射物体，因为所有方向都有相同的采样概率，并且没有对反射方向 t 的偏好。球面高斯不允许在优化的环境图中显示高频细节。最后，使用 IDE 在计算成本方面提供了优势：平均而言，场景优化需要 2.3 小时，而使用 TensoIR 则需要 5.0 小时。我们在图 12 中展示了一个具有均匀光滑材料的玩具示例，以及每列的粗糙度值直方图（顶部）。它显示了我们的模型能够捕捉更高频率的细节，同时不需要昂贵且不太准确的分层采样。

摘要

转移网络的消融。此外，我们评估了我们的转移网络（第 3.3 节）的变体，用于材料映射，增加其容量并添加残差连接。我们的研究表明，网络架构对性能影响不大，因为我们记录的法线 MAE 差异小于 1.9%，反照率 PSNR 差异小于 1.6%。这表明有限的容量足以学习 BRDF 变换。

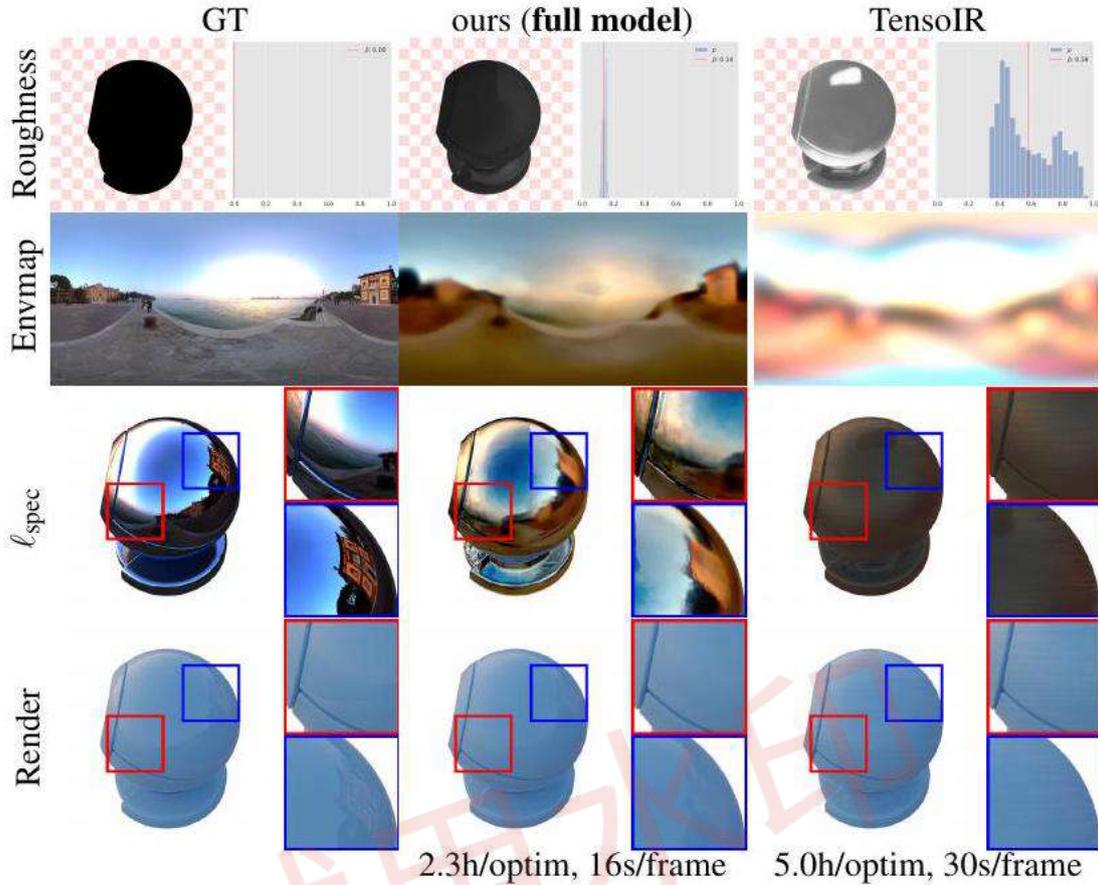


图 12: 与 TensoIR 的比较, TensoIR 使用球面高斯和光方向的分层采样来建模光照。此模型对应于我们消融中的 w/o g_{dir} , g_{indir} 。相反, 我们的方法使用神经表示来建模预积分光照。

4.4. 真实世界变换

在图 13 中, 我们定性地展示了我们的方法在我们的真实世界雕像数据集上的适用性。与 TensoIR 相比, 我们的分解更准确, 特别是在粗糙度方面, TensoIR 过饱和 (前两个示例), 以及反照率, TensoIR 倾向于变暗 (后两个示例)。总之, 这导致我们的渲染比 TensoIR 更逼真, 更接近参考图像。虽然我们注意到有改进的空间, 但我们强调我们的方法在相对不受控的设置下实现了可信的结果: 手持相机, 可能在不同的光照条件下捕获, 非线性相机 ISP, 以及相机姿态估计中的错误。这表明我们的方法对这些潜在的扰动具有有效的鲁棒性, 表明我们能够从一个雕像学习材料转移并将其真实地应用于另一个。

5. 局限性与结论

我们现在讨论局限性和扩展我们方法的途径。

局限性。虽然采用的预积分神经光模型优化速度快，但它不允许重新点亮场景，因为这需要重新训练模型。另一个局限性是我们的方法无法处理硬投影阴影；这些阴影最终会烘焙到反照率中，这与所有基线相同。一个解决方案可能是显式地结合遮挡估计。此外，我们的变换仅适用于与源场景相似的材料。为了扩展 \mathcal{F} 的分布域，最好从多个场景中同时学习变换，而不仅仅是一个。

问题设置。当前任务非常不受约束，估计



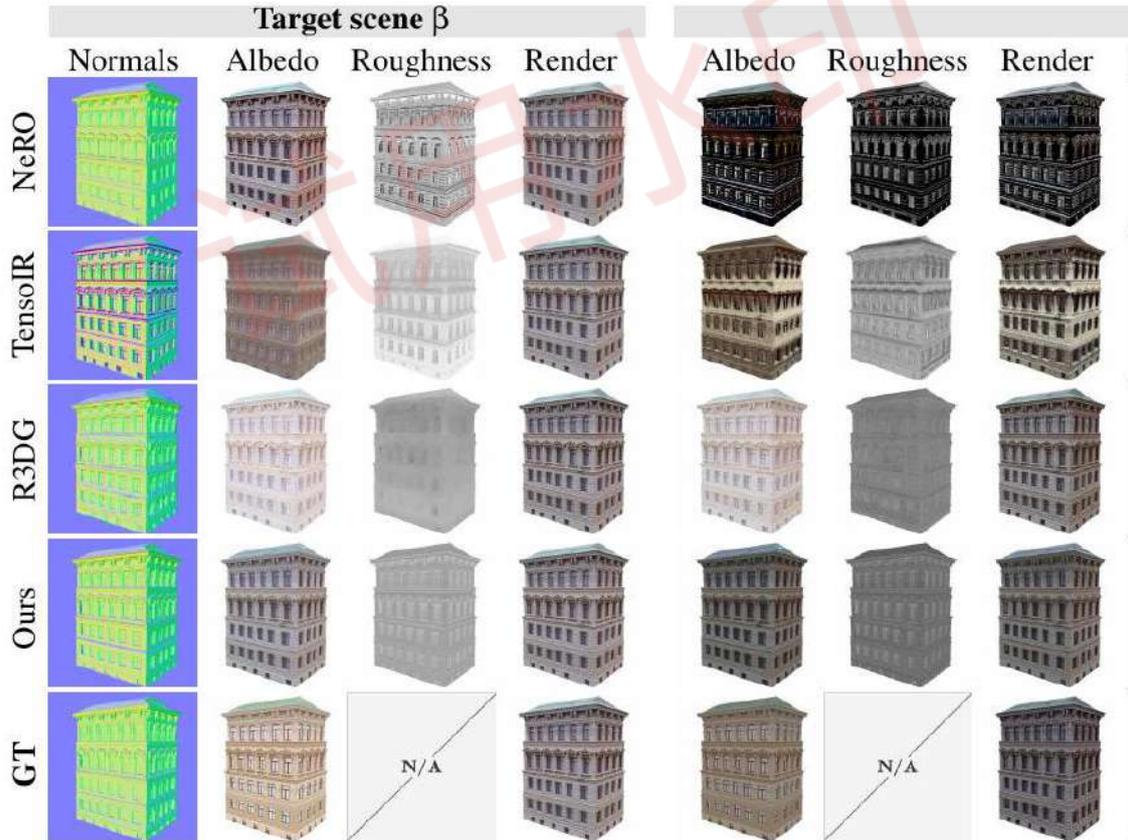
图 13: 真实数据上的定性材料转移。我们首先从一个用两种不同材料 (s_0 和 s_1 , 左) 捕获的雕像中学习材料转移函数。然后将学习到的变换应用于一个新雕像 (目标 s , 右), 显示估计的法线、反照率和粗糙度。最后, 将渲染的对象与参考照

片（最右）进行比较。我们提供了 TensoIR 和我们方法的结果。

从多视图输入中估计内在参数本身已经很具挑战性；在这里，我们旨在从未对齐的图像中端到端学习 BRDF 映射。不仅 s_0 和 s_1 的材料需要正确优化，目标场景也应该正确优化，因为不正确的几何估计将不可避免地导致不精确的材料变换估计。为了改进 \mathcal{F} 的估计，一个方向是强制更受控的环境，例如提供场景几何或施加固定光照。

扩展。一个有吸引力的途径是研究更丰富的材料变换。目前，我们假设函数仅依赖于 BRDF 参数，并且是逐点的，因此无法建模空间变化的变换。此外，变换是均匀的，即每个具有相同 β 的网格点将导致唯一的 $\mathcal{F}(\beta)$ 。这并不总是如此，例如对于湿度，朝上的表面可能比朝下的表面受影响更大。引入额外的条件到 MLP 建模 \mathcal{F} 或空间变化的 α 可以对此进行建模。另一个有趣的扩展是考虑时间变化的变换，如最近的工作 [NSO24]。

在本文中，我们引入了一个具有挑战性的材料变换估计任务。我们提出的解决方案允许从同一场景的两个观察中学习，具有单一联合优化表示。所展示的实验表明，学习到的变换可以转移到新场景。我们希望这将激发在这一方向上的新研究。



变换后的目标场景 $\mathcal{F}(\beta)$

反照率	粗糙度	Render	反照率	粗糙度
		<p>Render</p> 		
				
				
				

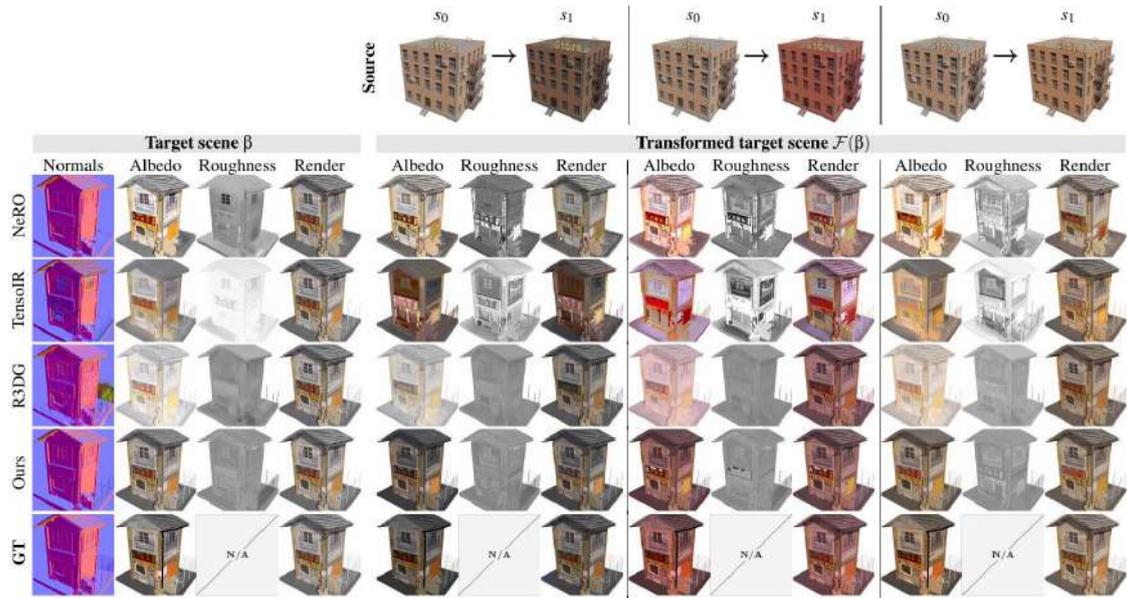


图 14: 定性材料变换结果。我们展示了使用学习到的变换函数 \mathcal{F} 合成新视图时的定性结果。对于每个子图，我们在顶部行显示源场景 (s_0, s_1) 上观察到的变换，具有三种可能的变换: T_1, T_2 和 T_3 列式。在左侧，我们显示目标场景的优化结果，在右侧，显示相应的三个源变换下的变换 BRDF。

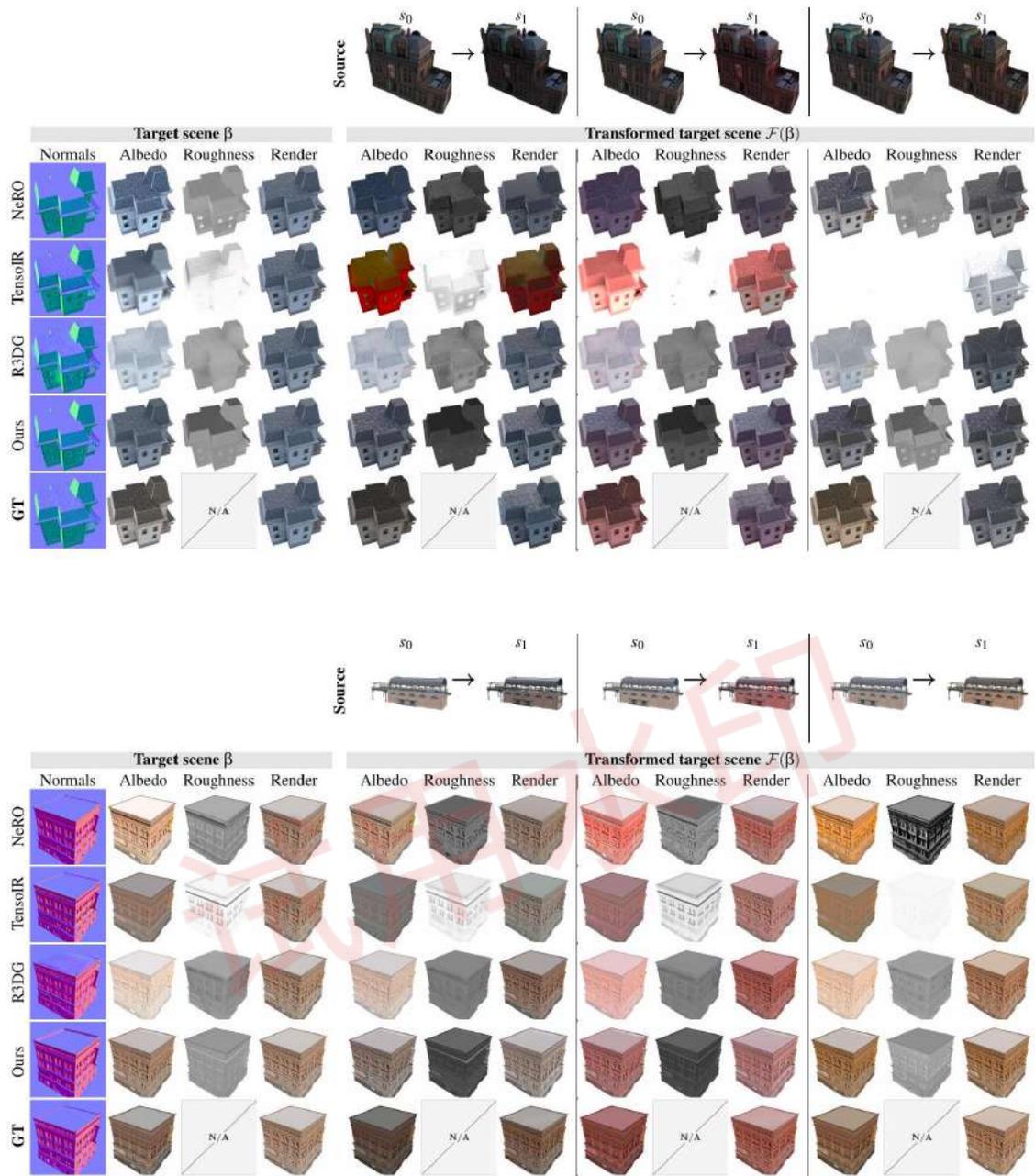


图 15: 定性材料变换结果。我们展示了使用学习到的变换函数 \mathcal{F} 合成新视图时的定性结果。对于每个子图，我们在顶部行显示源场景 (s_0, s_1) 上观察到的变换，具有三种可能的变换: T_1, T_2 和 T_3 列式。在左侧，我们显示目标场景的优化结果，在右侧，显示相应的三个源变换下的变换 BRDF。

Material transforms from disentangled NeRF representations

Ivan Lopes¹ Jean-François Lalonde² Raoul de Charette¹
¹Inria ²Université Laval

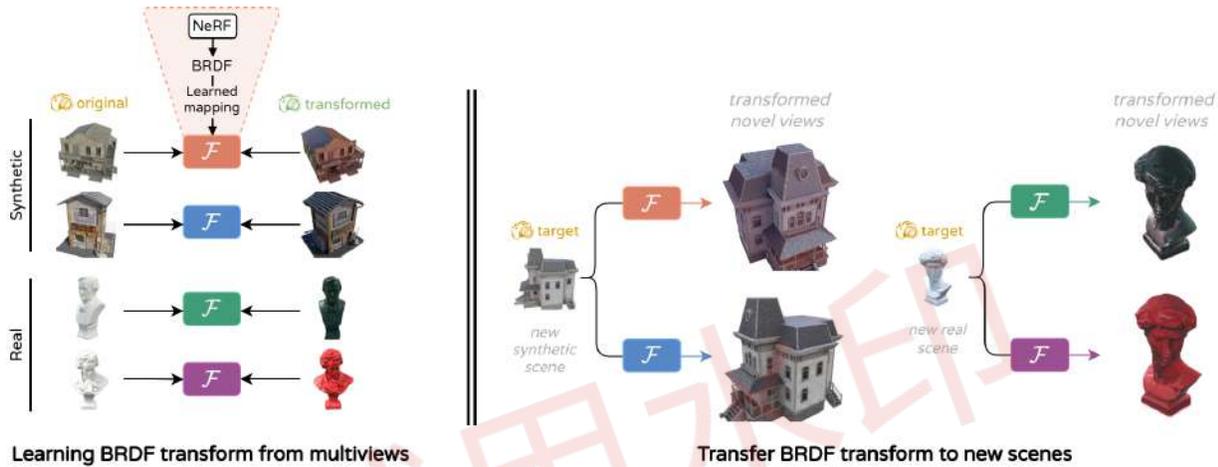


Figure 1: Proposed method. We illustrate our approach for inferring unknown material transformations in complex scenes. From a set of observations of a scene in two conditions: *original* and *transformed*, we leverage a joint Neural Radiance Field (NeRF) optimization to learn a material mapping function \mathcal{F} which models the observed changes at the material level accurately (e.g. the topmost transform on the left is a red varnish). This learned function can be applied to new *target* scenes with different geometry and material properties (right).

Abstract

In this paper, we first propose a novel method for transferring material transformations across different scenes. Building on disentangled Neural Radiance Field (NeRF) representations, our approach learns to map Bidirectional Reflectance Distribution Functions (BRDF) from pairs of scenes observed in varying conditions, such as dry and wet. The learned transformations can then be applied to unseen scenes with similar materials, therefore effectively rendering the transformation learned with an arbitrary level of intensity. Extensive experiments on synthetic scenes and real-world objects validate the effectiveness of our approach, showing that it can learn various transformations such as wetness, painting, coating, etc. Our results highlight not only the versatility of our method but also its potential for practical applications in computer graphics. We publish our method implementation, along with our synthetic/real datasets on <https://github.com/astra-vision/BRDFTransform>

1. Introduction

In computer graphics and vision, inverse rendering is key to extracting material information and allowing re-rendering under novel conditions (viewpoint, lighting, materials, etc.). While neural representations have largely taken over the traditional Physically-Based Rendering (PBR) techniques, recent works have demonstrated that the two representations can be combined [JLX*23], thus preserving the editability and expressivity of PBR representations along with the flexibility of neural representations.

When considering the appearance of a scene, certain transformations (such as applying a coat of varnish) can alter the material properties significantly, causing the scene’s appearance to change drastically. Currently, estimating the PBR characteristics of a known material after such a transformation requires capturing the scene again in the desired target condition. This process is both complex and laborious due to the variety of possible transformations, such as wetness, dust, varnish, painting, etc. In this work, we aim to learn a BRDF transformation from a source scene and apply it to different scenes.

Assuming we have paired observations of the same scene under two different conditions, say *original* and *varnished*, we propose a method to learn the transformation of materials. This transformation can then be applied to another scene composed of similar materials. This allows us to predict the appearance of that scene under this effect, effectively transferring the material transformation.

Fig. 1 illustrates that several material transformations can be learned from multiple pairs of scenes (left) and later applied on novel scenes (right), whether synthetic or real. Technically, our method relies on the joint optimization of a radiance field corresponding to a first scene captured in original and transformed (e.g., varnished) conditions, possibly with varying lighting conditions. We rely here on the disentangled NeRF representation of TensoIR [JLX*23]—that optimizes appearance, geometry, and parametric BRDF simultaneously—while introducing two novel key components. First, we condition the transformed scene BRDF on the original scene and approximate its transformation with a Multi-Layer Perceptron (MLP). Second, we expose a limitation of TensoIR showing it fails at decomposing highly reflective materials and propose an improved light estimation scheme that better estimates low roughness components while preserving high frequencies in the illumination. As a result, our framework allows capturing a collection of transformations which can then be applied on new scenes, while controlling the intensity of the transformation. We demonstrate the performance of our method on two new datasets: a synthetic dataset with a series of custom shader transformations and a real-world dataset of figurines with varying material conditions (e.g., original, painted, varnished, etc.). On both datasets, our approach produces faithful transformations. Our method and datasets will be released publicly.

2. Related work

Inverse rendering is a long-standing problem, it has gained interest recently with the use of neural radiance fields [MST*20]. Given a set of images of an object taken from different points of view, the goal is to optimize an implicit volumetric model for opacity and radiance. This allows synthesizing frames at novel viewpoints using volume rendering. Many works have extended this approach to learning a more explicit volume in which material information is disentangled from light sources. This way, the scene can be relit and the material manipulated, providing much more control over conventional radiance-centered methods.

BRDF estimation in NeRF. NeRD [BBJ*21] is the first method to perform BRDF optimization of a scene in an uncontrolled setting. Later, approaches such as NeRV [SDZ*21] and IndiSG [ZSH*22] introduced solutions for self-occlusions and indirect light. Spherical Gaussians (SG) have been widely used for modeling illumination in inverse rendering [ZLW*21, ZSH*22, JLX*23, ZXY*23]. Implicit representations were subsequently introduced in NeILF++ [YZL*22, ZYL*23], NeRO [LWL*23] and TensoSDF [LWZW24] to better represent high frequency illumination. For specular objects, some have proposed new forms of encodings to help supervise narrow specular lobes. For example, RefNeRF [VHM*22] uses the Integrated Directional Encoding (IDE), NeAI [ZZW*24] an Integrated Lobe Encoding (ILE), and SpecNeRF [MAT*24] Gaussian directional encodings. These optimiza-

tion methods have been combined with Signed Distance Functions (SDF) in Factored-NeuS [FSV*23] and NeRO [LWL*23] to provide a more robust geometry estimation. Recently, NeP [WHZL24] uses a neural plenoptic function to model incoming light. Unlike others who adopted analytical BRDFs, NeRFactor [ZSD*21] uses a data-driven approach by first learning priors on real-world BRDFs from the MERL dataset [MPBM03]. Instead, ENVIDR [LCL*23] learns this prior on a synthetic dataset. NVDiffrec/MC [MHS*22, HHM22] optimize the mesh and its materials as SVBRDF maps.

Closest to our approach in terms of scene optimization, TensoIR [JLX*23] adopts a tensor representation and factorizes a light component to learn under multiple illumination. They use stratified sampling and SGs to model direct light, while we adopt a neural representation. NeRO [LWL*23] has the same illumination approach but uses a two-stage approach which is computationally expensive. Instead, we use an approximation of the rendering equation to pre-compute part of the integral.

Material and neural transforms. While the problem of BRDF transform in a multi-view setting has not been explored, to the best of our knowledge, we present relevant research on this topic. In tvBRDF [SSR*07], the authors propose analytical models for transforms such as dust, watercolors, oils, and sprays, on non-spatially varying materials. Another line of research looks at translating a NeRF reconstruction based on an exemplar-style image. This includes StyleNeRF [LZC*23], LAENeRF [RSKS24], or iNeRF2NeRF [HTE*23] which is prompt-based. Also related is the task of performing material transfers such as in NeRF-analogies [FLNP*24]. In Climate-NeRF [LLF*23], global effects are injected into the scene but do not affect the object materials.

Inverse rendering datasets. Datasets with varying BRDFs were introduced in [GTR*06] with time-varying effects. They record a number of surfaces transformed by natural processes showing how it affects the BRDF temporally and spatially. It is common that inverse rendering datasets offer captures under different illuminations but the material remains unchanged: ReNe [TMS*23] proposes a dataset of 20 real scenes captured under 40 point-light positions. Objects-with-Lighting [UAS*24] introduces 8 objects under 3 environments with the corresponding High Dynamic Range (HDR) environment maps.

3. Method

3.1. Problem setting

Consider the scenario shown in Fig. 2, where a scene is observed in its *original* state s_0 , and a second time s_1 with its materials *transformed* by an unknown effect T , such that $s_1 = T(s_0)$. For example, T could be the result of applying a coat of paint, some colored varnish, or having the scene soaked with water. Note that s_1 might have been captured under a different illumination than s_0 . Our goal is to model the material transformation happening between s_0 and s_1 , in such a way that we can transfer this effect to a new scene.

We model the scene with a BRDF field, more specifically, every point of a scene s is characterized by material properties $\beta = (\rho, r) \in \mathbb{R}^4$, where ρ is the albedo (in RGB) and r the roughness. Our formulation assumes that the original scene s_0 is affected

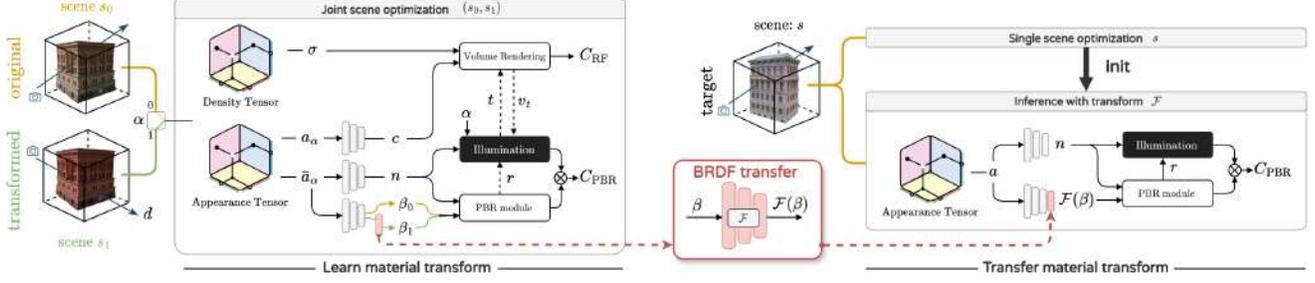


Figure 2: Overview of our proposed method. Our method takes observations of the same scene with two different materials (β_0, β_1) for s_0 and s_1 , respectively. We assume β_1 to be a function of β_0 . Our method learns a joint representation and a transform function \mathcal{F} which maps the material of the first to the second (left block). Given a new scene s , we learn its geometry and material and apply our learned transform function (right block) to produce the same effects observed in the source scenes (s_0, s_1) .

by an unknown transformation which changes its material properties β_0 but not its geometry, resulting in scene s_1 with material β_1 . Our method aims to learn a function \mathcal{F} which approximates the unknown mapping T between the two materials β_0 and β_1 , in such a way that \mathcal{F} can be applied on new scenes as shown in Fig. 2 (right).

3.2. Preliminaries

Our optimization approach is based on TensoIR [JLX*23], itself derived from TensoRF [CXG*22], to learn a neural radiance field of the scene. For clarity, we follow their notation here. In this framework, a radiance field is learned by jointly training both a density tensor \mathcal{G}_σ and an appearance tensor \mathcal{G}_a . From the latter, surface normals n and material properties β can be estimated at every 3D point x using lightweight MLPs, noted \mathcal{D} , and accumulated along each viewing rays using volume rendering. While the scene can be imaged under a single illumination condition, TensoIR also supports multiple observations of the scene under different illuminations. In this case, it further factorizes a light embedding to produce light-dependent appearance features a_α , where α indexes the lighting conditions (modeled as an environment map). The estimated quantities at every point x of the scene can therefore be written as:

$$n = \mathcal{D}_n(\bar{a}_\alpha), \quad \beta = \mathcal{D}_\beta(\bar{a}_\alpha), \quad c_\alpha = \mathcal{D}_c(a_\alpha), \quad (1)$$

where \bar{a}_α is the average appearance features across both light embeddings, and c is the pixel color (as in the original TensoRF formulation). TensoIR learns a disentangled representation, allowing the color of each point x to be estimated for a given view direction d either through volume rendering, represented as $C_{RF}(x, d)$, or through physically-based rendering, also represented as $C_{PBR}(x, d)$ — both of which are supervised by the reference images.

3.3. Learning material transforms

As discussed in sec. 3.1, we aim to learn \mathcal{F} which maps the BRDF parameters β_0 of a scene s_0 to its transformed appearance β_1 for s_1 . As illustrated in Fig. 2, we formulate the transfer with:

$$\beta_\alpha = \beta_0[\alpha = 0] + \mathcal{F}(\beta_0)[\alpha = 1] \quad (2)$$

where $[\bullet]$ is the Iverson bracket and \mathcal{F} is a small MLP network that is trained end-to-end together with the appearance and density

tensors. Here α is an indicator representing whether we are rendering the original scene (i.e. $\alpha = 0$) or its transformed version (i.e. $\alpha = 1$). Using this formulation, we jointly train on s_0 and s_1 , and learn a single neural representation for both scenes.

3.4. Light estimation

Limitation of TensoIR. We observe that the original TensoIR framework struggles in reconstructing low-roughness scenes (Fig. 3), which is crucial for representing glossy surfaces. We also note that the low number of spherical gaussians used to represent the environment results in the absence of high-frequency content in the lighting. The use of stratified sampling and low-frequency light representation comes at the cost of incorrect estimation of objects with low roughness.

To alleviate this problem and allow learning a wider variety of material transforms, we propose an improvement to the formulation by borrowing ideas from NeRO [LWL*23]. We keep the volume representation of TensoIR as it is fast to optimize but avoid expensive light sampling by following NeRO. That way, we benefit from both methods and ensure fast optimization speeds.

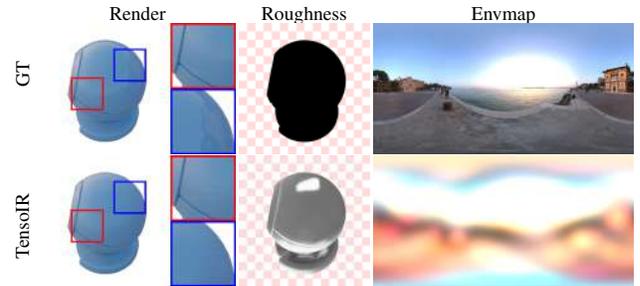


Figure 3: TensoIR on glossy surfaces. We observe that TensoIR overestimates roughness and smooths the estimated illumination.

Formulation. Rendering the color of a point x from a viewing direction d is given by

$$C_{PBR}(x, d) = \int_{\Omega} L(\omega, x) f_r(\omega, d) (\omega \cdot n) d\omega, \quad (3)$$

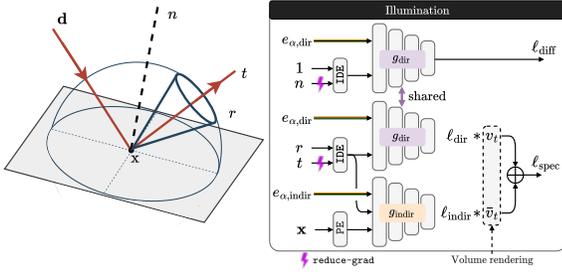


Figure 4: Light estimation. We adopt a neural light representation [LWL*23] which models direct and indirect light sources separately. On the indirect component, the two types of light sources are blended using an occlusion mask obtained via secondary ray casting along reflected light direction t [JLX*23]. To avoid disrupting the optimization of the geometry, we reduce the gradient intensity along the directional inputs (on n and t). We note $\bar{v}_t = 1 - v_t$, IDE is an Integrated Directional Encoding [VHM*22] while PE is a Positional Encoding [MST*20].

where Ω is the integrating hemisphere, L the light intensity from direction ω at x . Here, the BRDF f_r is parameterized with material properties $\beta = (\rho, r)$. We adopt the micro-facet reflectance model of [CT82]:

$$f_r(\omega, d) = \frac{\rho}{\pi} + \frac{DFG}{4(\omega \cdot n)(d \cdot n)}, \quad (4)$$

where D , F , and G are the normal distribution, Fresnel, and geometric attenuation terms. For brevity, we omit the parameters for these three functions. We follow NeRO [LWL*23] and use the split-sum approximation on the specular component [Kar13]. After integration, it becomes:

$$CPBR(x, d) = \rho \ell_{\text{diff}} + M_{\text{spec}} \ell_{\text{spec}}, \quad (5)$$

where

$$\ell_{\text{diff}} = \int_{\Omega} L(\omega, x) D(n, 1) d\omega, \quad \ell_{\text{spec}} = \int_{\Omega} L(\omega, x) D(t, r) d\omega, \quad (6)$$

and

$$M_{\text{spec}} = \int_{\Omega} \frac{DFG}{4(d \cdot n)} d\omega. \quad (7)$$

Here, t is the reflected direction w.r.t. surface normal n . Note that M_{spec} can be precomputed as it does not depend on L . The integrals ℓ_{diff} and ℓ_{spec} (which depend on L) are discussed next.

Light estimation. We use the Integrated Directional Encoding (IDE) of Ref-NeRF [VHM*22] to model the scene illumination. Similar to NeRO [LWL*23], we leverage two MLPs for approximating L , being g_{dir} for direct and g_{indir} for indirect (e.g., interreflections) light. To accommodate for a joint optimization setting on two scenes, we feed latent embeddings to both light MLPs g in order to account for possible changes in lighting. This is achieved by channel-wise concatenating the corresponding embedding to the IDE of the g inputs, depending on whether the original or transformed scene is rendered. The illumination expressions are written

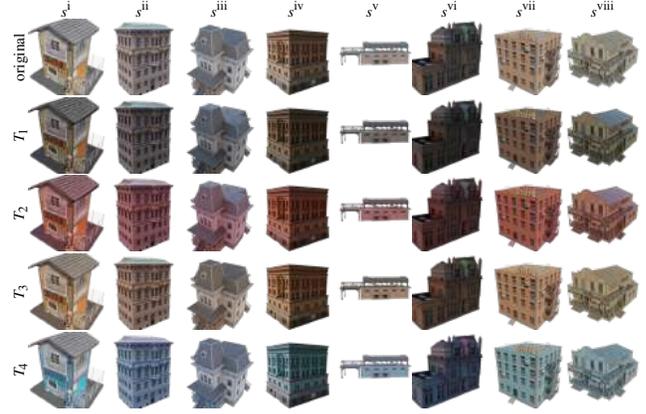


Figure 5: Synthetic dataset. Each column shows a difference scene $s^k, k \in \{i, \dots, \text{viii}\}$. The first row shows the original scene, each subsequent row shows the scene after each synthetic transformation $T_j, j \in \{1, \dots, 4\}$.

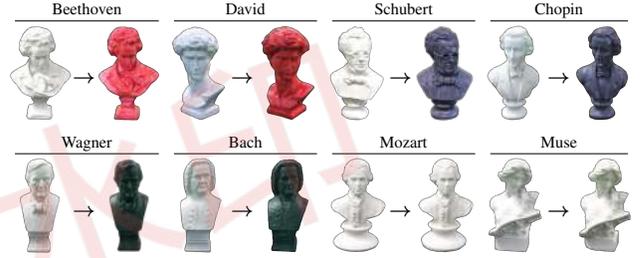


Figure 6: Real-world dataset. Different bust figurines were first photographed with and without various colored coats (Beethoven, David, Schubert, Chopin, Wagner, Bach) or glossy varnishes (Mozart, Muse).

as:

$$\begin{aligned} \ell_{\text{diff}} &= g_{\text{dir}}(\text{IDE}(n, 1), e_{\alpha, \text{dir}}), \\ \ell_{\text{spec}} &= v_t g_{\text{dir}}(\text{IDE}(t, r), e_{\alpha, \text{dir}}) \\ &\quad + (1 - v_t) g_{\text{indir}}(\text{IDE}(t, r), x, e_{\alpha, \text{indir}}). \end{aligned} \quad (8)$$

For the specular term ℓ_{spec} , v_t is the visibility term obtained by ray tracing along t in the density volume. A detail of the proposed illumination components is shown in Fig. 4. Here, the latent embeddings e provide conditioning to the g networks in order to best model scene-specific illuminations.

4. Experiments

4.1. Experimental methodology

Given the lack of available datasets suitable for studying BRDF transfer, we build two datasets: one synthetic with custom Blender shaders; and one real capturing figurines under varying material conditions. Both datasets will be shared publicly. Given the complexity of acquiring real-world ground truth decompositions, we follow the usual practice [JLX*23] and report quantitative perfor-

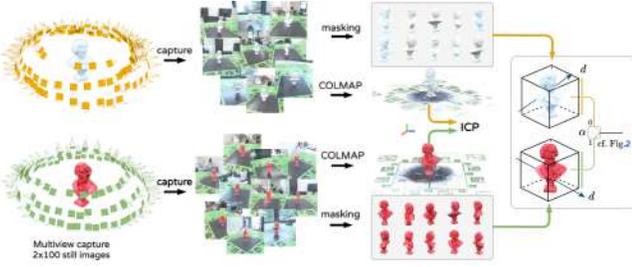


Figure 7: Capture and pre-processing of real data. From left to right, we start by capturing still images for each variant of the object s_0 (top) and s_1 (bottom). Then we apply COLMAP to retrieve the dense reconstruction of both sets. Using ICP, we align both shapes to set all camera poses on the same reference basis.

mance on the synthetic dataset only. Qualitative results are shown for both.

Synthetic dataset. We obtain eight freely available and open-source 3D models compatible with the Blender PBR rendering pipeline. Models are processed individually and rescaled so they share a similar size. We design a simple set of shader transforms $T \in \{T_1, \dots, T_n\}$ in Blender to alter all materials in a given scene. This allows us to control the global α (0: no change, 1: fully transformed) of a scene and the type of transformation applied. Following the dataset creation procedure of NeRFactor [ZSD*21], we render 100/20 training/test views for each scene. An overview of the dataset is provided in Fig. 5, which shows all scenes and synthetic transformations. Next, we briefly describe the different synthetic transformations:

- In ‘original’, the original PBR materials are used;
- T_1 : $r' = 0$ and ρ' has 30% the HSV value of ρ ;
- T_2 : $r' = 0$ and $\rho' = 0.5\rho + 0.5\rho_{\text{red}}$;
- T_3 : $r' = 1$ and $\rho' = 0.2\rho + 0.8\rho_{\text{sand}}$;
- T_4 : $r' = r$ and ρ' has the opposite hue of ρ .

where ρ_{sand} and ρ_{red} are two RGB colors chosen arbitrarily. While realistic transformation would require complex shaders, we highlight that our choice of transformations is motivated by visual approximation of real-world transformations, being: wetness (T_1), fresh painting (T_2), dustiness (T_3), painting (T_4).

Real-world dataset. We collected eight bust figurines (approximately 10 cm high, see Fig. 6) and photographed them all around using a phone. We captured their appearance in two different conditions: first in their original appearance, and once more after altering their material condition, for example by applying various colored coats or varnishes. Unlike synthetic data, capturing real-world data results in unknown camera poses and two misaligned sets of photographs since they cannot be taken from exactly the same viewpoint. This can be prevented using a specially-designed camera rig as in [TMS*23], but comes at the cost and time of building and calibrating the apparatus. Instead, and as shown in Fig. 7, we apply COLMAP [SF16, SZPF16] separately to the original and transformed sets of images and then estimate the rigid transformation matrix between the two resulting point clouds with the itera-

tive closest point (ICP) algorithm. The resulting matrix is applied to correct the camera poses from both sets into a unique reference basis. Our optimization requires that we mask out the scene backgrounds, to do so we employ the library rembg [Gat20] and manually correct the frames that are not masked properly.

Training. We train in mixed batches with pixel rays from both s_0 and s_1 . All components are optimized end-to-end. The optimization of the target scene, on which we apply the learned transformation is done without illumination embeddings e_α . To apply the learned transform function on a new scene, we simply plug-in our trained MLP and compute $\mathcal{F}(\beta)$, shown on the right in Fig. 2.

Network. To implement our approach described in sec. 3, we adopt the same base architecture and optimization procedure as TensorIR [JLX*23]. Vectors e_α are light embeddings of size 72 that are used to encode scene information that is specific to each α as both observations s_0 and s_1 might be captured under slightly different lighting conditions. As in TensorIR, we use secondary ray marching to estimate the visibility mask v_r , but instead of sampling the radiance field, we leverage a dedicated g_{indir} MLP to estimate the irradiance for occluded directions. During the backward pass, the `reduce-grad` function applies a 10^{-2} weight on the gradient in order to reduce its effect on directional inputs. The model used to learn the transform function \mathcal{F} is a small MLP with a single hidden layer of dimension 256.

Baselines. To the best of our knowledge, there are no image-based material transform learning methods. Therefore, we made our best efforts to build strong baselines from existing techniques.

First, we note that TensorIR [JLX*23] can be adapted by replacing the input of \mathcal{D}_β from \bar{a}_α to a_α for material-specific information. This however poses two problems. First, a_α is not interpretable since it contains entangled information corresponding to the geometry, material, and illumination; second, transferring this function to a new scene would fail as the appearance features from both scenes would belong to different embedding spaces.

We therefore choose a different approach to setup a fair baseline. We first train on the original scene s_0 and the transformed scene s_1 , separately. Then, we extract the geometry and BRDF from both scenes by querying the volume. A MLP model is trained to learn the mapping between the two sets of BRDF. Finally this model is applied on a new scene s , such as to map its material from β to $\mathcal{F}(\beta)$. We follow this for several methods: on the vanilla TensorIR itself [JLX*23] as well as on two recent inverse rendering methods: NeRO [LWL*23] and Relightable 3D Gaussians (RG3D) [GGL*23]. Since some baselines predict an additional metalness component while we do not, we set their values to zero during optimization and novel view synthesis to avoid having an unfair advantage.

4.2. Main results

Learning BRDF transfer. In Tab. 1 we present the BRDF transfer results averaged over all synthetic scenes for each transformation T , and the mean over all transformations. To assess the quality of our decomposition and material transfer, we report Mean Angular Error (MAE) for normals (\downarrow) and PSNR (\uparrow), SSIM (\uparrow), and

Transfer	Method	Normals*		Albedo		Render	
		MAE \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	
T_1	NeRO	7.726	16.03	0.676	0.254	20.40	
	TensoIR	10.92	17.82	0.722	0.262	21.04	
	R3DG	11.41	11.57	0.610	0.223	26.12	
	ours	6.750	19.80	0.781	0.195	21.95	
T_2	NeRO	7.726	16.54	0.693	0.242	21.47	
	TensoIR	10.92	16.66	0.688	0.251	20.02	
	R3DG	11.41	12.44	0.635	0.219	27.36	
	ours	6.750	18.62	0.770	0.199	22.44	
T_3	NeRO	7.726	16.21	0.690	0.248	24.00	
	TensoIR	10.92	16.90	0.697	0.293	23.72	
	R3DG	11.41	12.79	0.656	0.217	31.45	
	ours	6.750	19.81	0.787	0.197	29.76	
T_4	NeRO	7.726	15.43	0.662	0.263	22.60	
	TensoIR	10.92	17.52	0.696	0.273	22.61	
	R3DG	11.41	13.26	0.668	0.217	29.12	
	ours	6.750	18.88	0.766	0.203	26.88	
mean	NeRO	7.726	16.05	0.680	0.252	22.12	
	TensoIR	10.92	17.23	0.701	0.270	21.85	
	R3DG	11.41	12.52	0.642	0.219	28.51	
	ours	6.750	19.28	0.776	0.199	25.26	

* Normals are independent of the transformation learned.

Table 1: Novel view transfer evaluation. We evaluate the material estimation on our synthetic dataset by measuring metrics after transferring cross-scenes. Evaluation on the test set on novel view synthesis. We highlight **best** and **2nd best**.

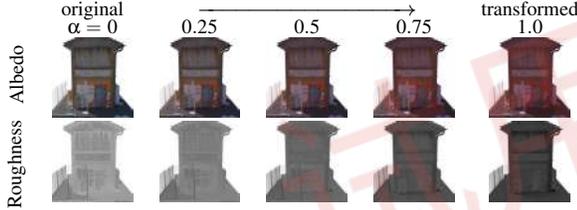


Figure 8: Transform interpolation. We linearly interpolate the BRDF parameters between the original and target scene for varying values of $\alpha \in [0, 1]$.

LPIPS (\downarrow) for the estimated albedo. For completeness, we also report PSNR (\uparrow) of the rendering for novel view synthesis. Results in the table advocate that our method better estimates normals and albedo, showcasing that it faithfully learns the transformation of the BRDF function. Importantly, note that R3DG have higher fidelity renderings although this comes at the cost of inaccurate decomposition (normals, albedo) which suggests entangled material information in the scene lighting. Corresponding qualitative results are shown in Fig. 14 and Fig. 15 for four scenes and three transformations. The latter demonstrates the superiority of our method, producing faithful renderings without compromising the scene decomposition (normals, albedo, roughness). In Fig. 8 we further show that our formulation allows interpolation between the original ($\alpha = 0$) and the learned transformed BRDF ($\alpha = 1$), according to eq. (2).

Additionally, we provide heatmaps of per-scene Albedo performance in Fig. 9 for all transformation, offering a more fine-grained analysis. Overall, heatmaps indicate that some scenes are easier to

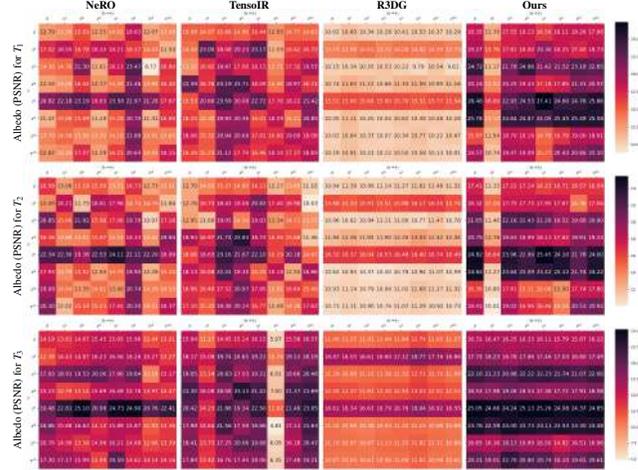


Figure 9: Performance of BRDF transfer per transformation. We provide heatmaps of Albedo PSNR (\uparrow) for pairs of source (horizontal) and target (vertical) scenes. The diagonal indicates performance of the BRDF transformation when applied on the same scene. Of note, some scenes are easily transformed (s^i , s^v), arguably because of simpler appearance (cf. Fig. 5) while TensoIR seems to struggle to learn T_3 on the s^v scene. Despite great rendering capability (cf. Tab. 1), R3DG struggles to faithfully decompose the scene, while our method consistently outperforms all baselines.

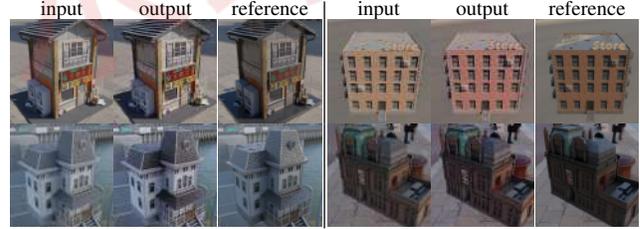


Figure 10: Examples of prompt-driven translation. Instruct-Pix2Pix [BHE23] applied on images (with background) with the input prompt “make it more glossy”. Image translation methods limit the consistency of the output in terms of geometry and appearance. It also assumes the transformation is known and can be formulated as a prompt, which is not always the case.

transform, such as s^i and s^v (cf. Fig. 5 for reference), whereas learning from s^v proved to be more complex – especially for TensoIR with transformation T_3 . As shown in Tab. 1, R3DG struggles to decompose the albedo correctly, while our method outperforms all baselines in almost all instances.

Comparison to i2i translation methods. Additionally, we provide in Fig. 10 some results with the prompt guided image-to-image (i2i) translation method InstructPix2Pix [BHE23]. The result exhibits the limitation of such techniques which require a priori knowledge about the transformation and its formulation with natural language. Also, the output is not geometrically consistent.

4.3. Ablation study

We conduct our ablations on the synthetic dataset of sec. 4.2 because of the availability of ground truth material properties. As before, we report metrics on estimated albedo and novel view synthesis. Here, we name our proposed method “full model”; “w/o reduce-grad” allows the gradient to flow on the directional inputs of the illumination module without damping; “w/o g_{dir}, g_{indir} ” removes the illumination MLPs and uses spherical gaussians and stratified sampling to represent light sources (as in TensoIR); “w/o joint optim.” corresponds to learning both scene representations separately and fitting a MLP to learn \mathcal{F} ; “w/o transfer” acts as the lower bound with metrics of the original scene s_0 computed against the reference images of s_1 .

Impact of design choices on the transfer capability. In Tab. 2, we present an evaluation of the components on the ability to transfer the learned transformation to other scenes. Using a neural light representation improves the geometry estimation as well as the overall quality of the predicted albedo. We note that learning on both the original and transformed scenes benefits the transfer of \mathcal{F} .

Ablations	Normals		Albedo		Render
	MAE $_{\downarrow}$	PSNR $_{\uparrow}$	SSIM $_{\uparrow}$	LPIPS $_{\downarrow}$	PSNR $_{\uparrow}$
ours (full model)	6.750	19.80	<u>0.781</u>	<u>0.195</u>	21.95
↳ w/o g_{dir}, g_{indir}	<u>9.060</u>	<u>18.51</u>	0.784	0.187	<u>21.04</u>
↳ w/o joint optim.	11.14	17.75	0.714	0.264	20.50

Table 2: Transfer to other scenes. Evaluates the transform T_1 , on novel view synthesis after applying \mathcal{F} on a new scene s_0 . The resulting scene is evaluated against the reference images of s_1 , which were rendered with the ground truth transform T_1 .

Benefit of joint training. It is also interesting to study the impact of each component when applying the learned BRDF transform to the *same scene* (rather than a new scene as done previously). This allows to evaluate the quality of the BRDF transform learned on the source scene. Results from this experiment are presented in Tab. 3. This shows the benefit of learning the transform function during optimization of the scene. We remark a slightly better MAE on “w/o transfer” as the geometry is learned on the original scene which is often easier to estimate compared to the transformed scene.

Ablations	Normals		Albedo		Render
	MAE $_{\downarrow}$	PSNR $_{\uparrow}$	SSIM $_{\uparrow}$	LPIPS $_{\downarrow}$	PSNR $_{\uparrow}$
ours (full model)	<u>7.164</u>	<u>21.40</u>	<u>0.805</u>	<u>0.187</u>	30.51
↳ w/o g_{dir}, g_{indir}	9.338	23.80	0.851	0.210	<u>29.58</u>
↳ w/o joint optim.	11.14	19.43	0.747	0.242	22.37
↳ w/o transfer	6.750	18.56	0.769	0.175	21.12

Table 3: Transformation of the same scene. We measure the gain related to optimizing jointly s_0 and s_1 and ablate the different components. In “w/o joint optim.” the two scenes are optimized separately, while the last experiment “w/o transfer” is by evaluating straight from s_0 with no transfer.

Effect of reduce-grad. When adopting integrated directional encoding (IDE) for the illumination components, we notice that the normals tend to degrade. Considering the shorter gradient path, it is

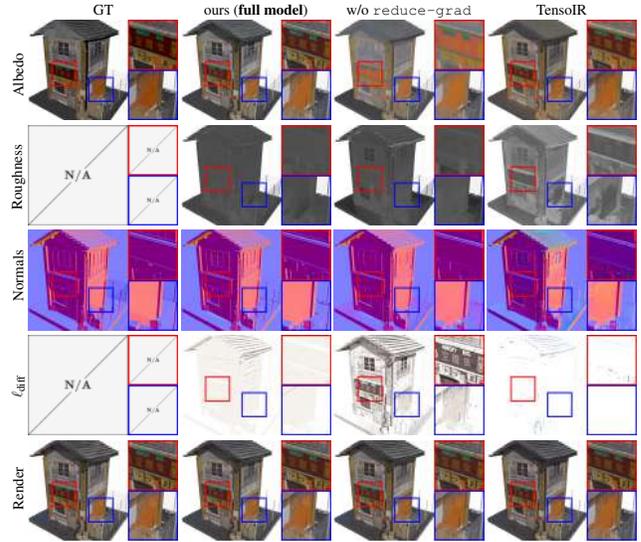


Figure 11: Illumination ablation. We show a detailed breakdown of the scene s_1^i , we notice that when learning from two scenes with a neural light representation, the diffuse light ℓ_{diff} tends to overfit to the geometry of the scene which leads to color information leaking from the albedo into the light. In contrast, reducing the gradient on directional input n and t of g_{dir} alleviates this effect resulting in a uniform diffuse light.

much easier for the light MLP to bake albedo information at the expense of worsening surface normals and albedo. We can see this in the “w/o reduce-grad” column of Fig. 11: the billboard (cf. red zoom-in region) has high-frequency details baked into light while it is supposed to be a perfectly flat surface. Damping the gradient with the proposed reduce-grad operator prevents the normals from overfitting to the light gradient signal and results in a more uniform diffuse light estimation ℓ_{diff} .

Comparison to TensoIR. The illumination used in TensoIR [JLX*23] revolves around stratified sampling which doesn’t allow rendering low roughness surfaces. As such it is not capable of modeling reflective objects since all directions have the same probability of being sampled and there is no preference over the direction of reflection t . Spherical Gaussians do not allow for high-frequency details in the optimized environment map. Finally, using IDE provides an edge in terms of computation cost: on average, a scene optimization takes 2.3 hours, compared to 5.0 hours with TensoIR. We show in Fig. 12 a toy example with uniform glossy material along with a histogram of roughness values for each column (top). It shows that our model is able to capture higher frequency details while not requiring expensive and less accurate stratified sampling.

Ablations of the transfer network. Further, we evaluated variations of our transfer network (sec. 3.3) for material mapping, increasing its capacity and adding residual connections. Our findings indicate that the network architecture has little effect on the performance, as we recorded less than 1.9% difference in Normals MAE and 1.6% in Albedo PSNR. This suggests that limited capacity is sufficient for learning BRDF a transformation.

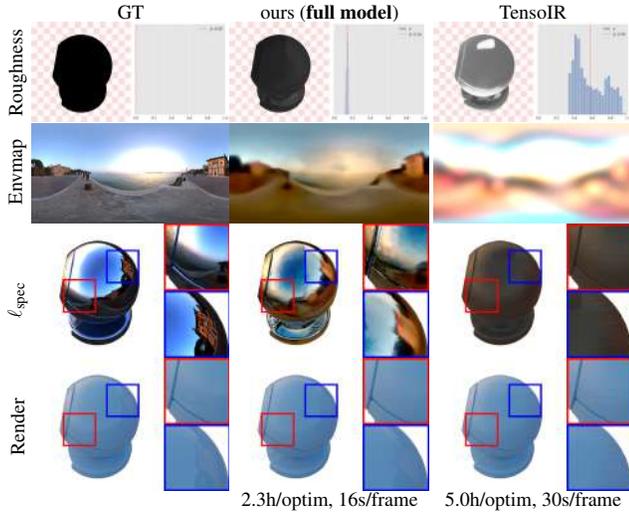


Figure 12: Comparison to TensoIR, which models illumination with spherical Gaussians and stratified sampling of the light directions. This model corresponds to w/o g_{dir}, g_{indir} in our ablations. Instead, our method uses a neural representation to model pre-integrated illumination.

4.4. Real world transformations

In Fig. 13, we qualitatively demonstrate the applicability of our method on our real-world figurines dataset. Compared to TensoIR, our decompositions are more accurate, particularly in terms of roughness, which is oversaturated by TensoIR (top two examples), and albedo, which TensoIR tends to darken (bottom two examples). Altogether, this leads to our renderings being more realistic than TensoIR and more closing resembling the reference images. While we denote margin for improvement, we highlight that our method achieves believable results despite data being captured in relatively uncontrolled settings: handheld camera, possibly varying illumination conditions across captures, non-linear camera ISP, and errors in camera pose estimation. This demonstrates our method’s effective robustness to these potential perturbations, showing that we are able to learn a material transfer from one figurine and apply it realistically to another.

5. Limitations & Conclusions

We now discuss limitations and avenues for extending our method.

Limitations. Although the adopted pre-integrated neural light model is fast to optimize, it does not allow relighting the scene, as this would require retraining the model. Another limitation is that our method cannot handle hard-cast shadows; these shadows end up baked into the albedo, as is the case with all baselines. One solution could be to explicitly incorporate an occlusion estimation. Additionally, our transformation is only applicable to materials that resemble those of the source scene. To expand the distribution domain of \mathcal{F} , it would be beneficial to learn the transformation from multiple scenes at once instead of just one.

Problem setting. The current task is very underconstrained, esti-

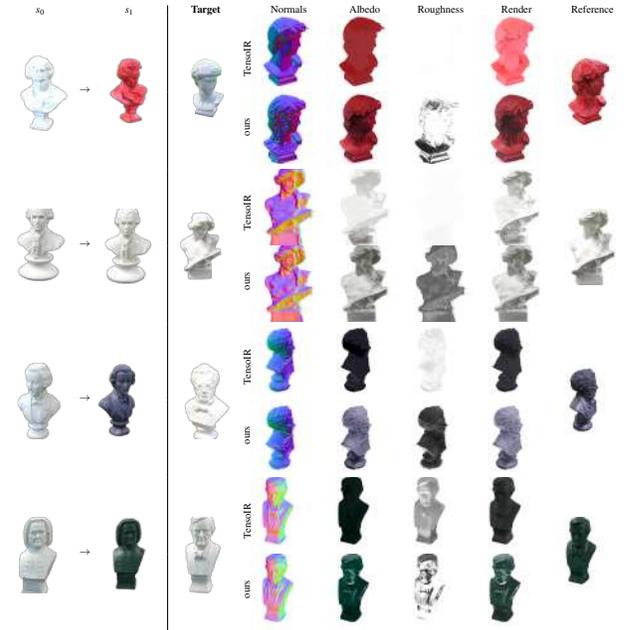


Figure 13: Qualitative material transfers on real data. We first learn the material transfer function from a figurine captured with two different materials (s_0 and s_1 , left). The learned transformation is then applied to a new figurine (target s , right), with the estimated normals, albedo, and roughness shown. Finally, the rendered object is compared to the reference photograph (far right). We provide results for TensoIR and our method.

imating intrinsic parameters from multi-view inputs is already challenging in and of itself; here we aim to learn a BRDF mapping end to end from unaligned images. Not only the materials of s_0 and s_1 need to be correctly optimized, but the target scene should also be properly optimized since improper geometry estimation would inevitably lead to estimation of imprecise material transforms. For improved estimation of \mathcal{F} , a direction is to enforce a more controlled environment such as providing the scene geometry or imposing a fixed illumination.

Extensions. An appealing avenue would be to work on richer material transformations. Currently, we assume the function to depend on the BRDF parameters alone and is point-wise, so there is no way to model spatially varying transformations. Furthermore, the transformation is uniform, *i.e.* every point of the mesh with identical β will result in a unique $\mathcal{F}(\beta)$. This is not always the case, for example for wetness, surfaces pointing upwards might be more affected than those pointing downwards. Introducing additional conditioning to the MLP modeling \mathcal{F} or a spatially varying α could model that. Another interesting extension is to consider time-varying transformation such as in recent works by [NSO24].

In this paper, we have introduced a challenging task of material transform estimation. Our proposed solution allows learning from two observations of the same scene with a single jointly optimized representation. The presented experiments demonstrate that the learned transformation can be transferred to new scenes. We hope this will motivate new research in this direction.

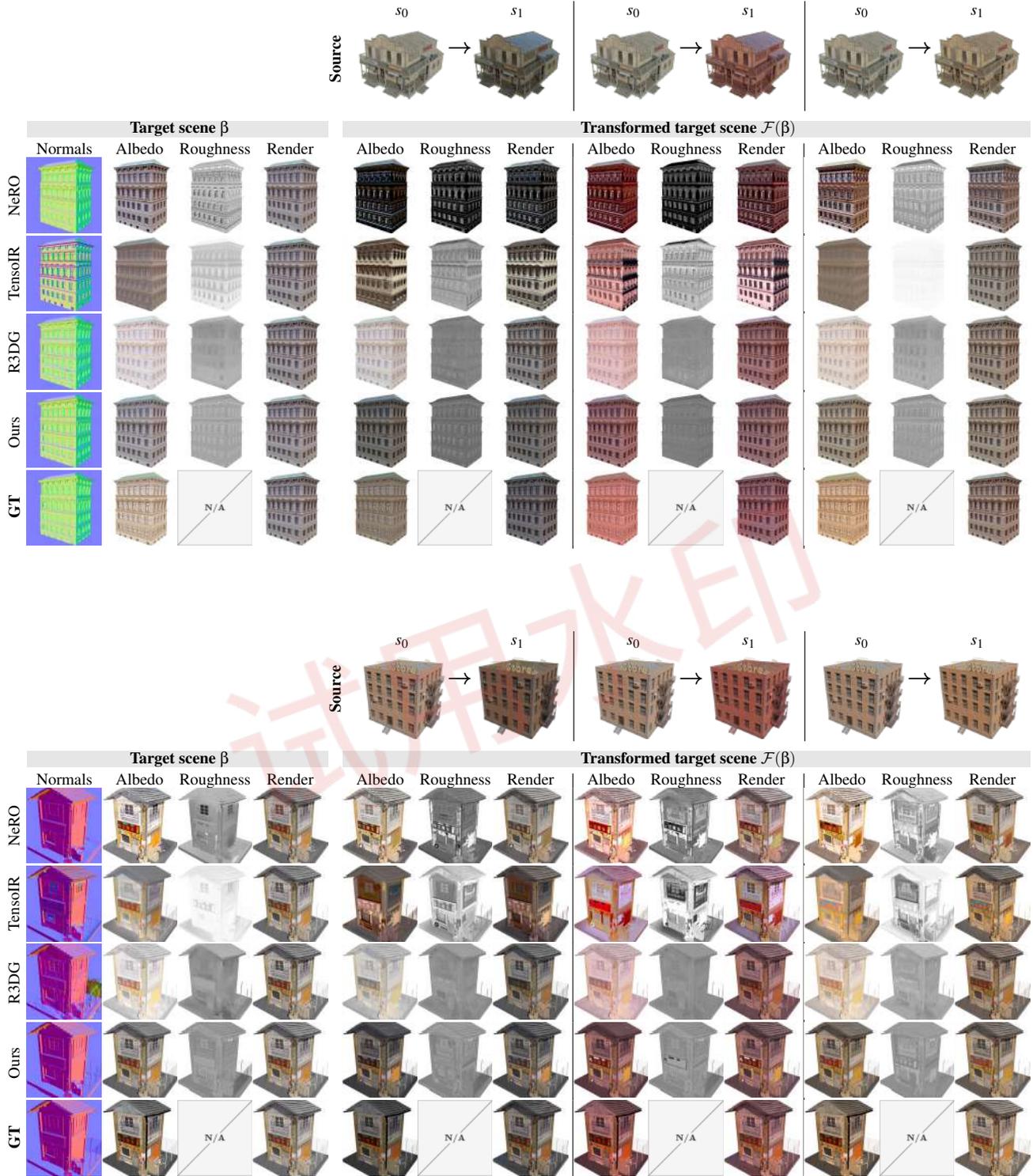


Figure 14: Qualitative material transforms results. We show qualitative results when synthesizing novel views with the learned transform function \mathcal{F} . For each sub-figure, we show in the top row the observed transform on the source scenes (s_0, s_1), with three possible transformations: T_1, T_2 , and T_3 column-wise. On the left, we show the optimization results of the target scene, and on the right, the transformed BRDF below the corresponding three source transforms.

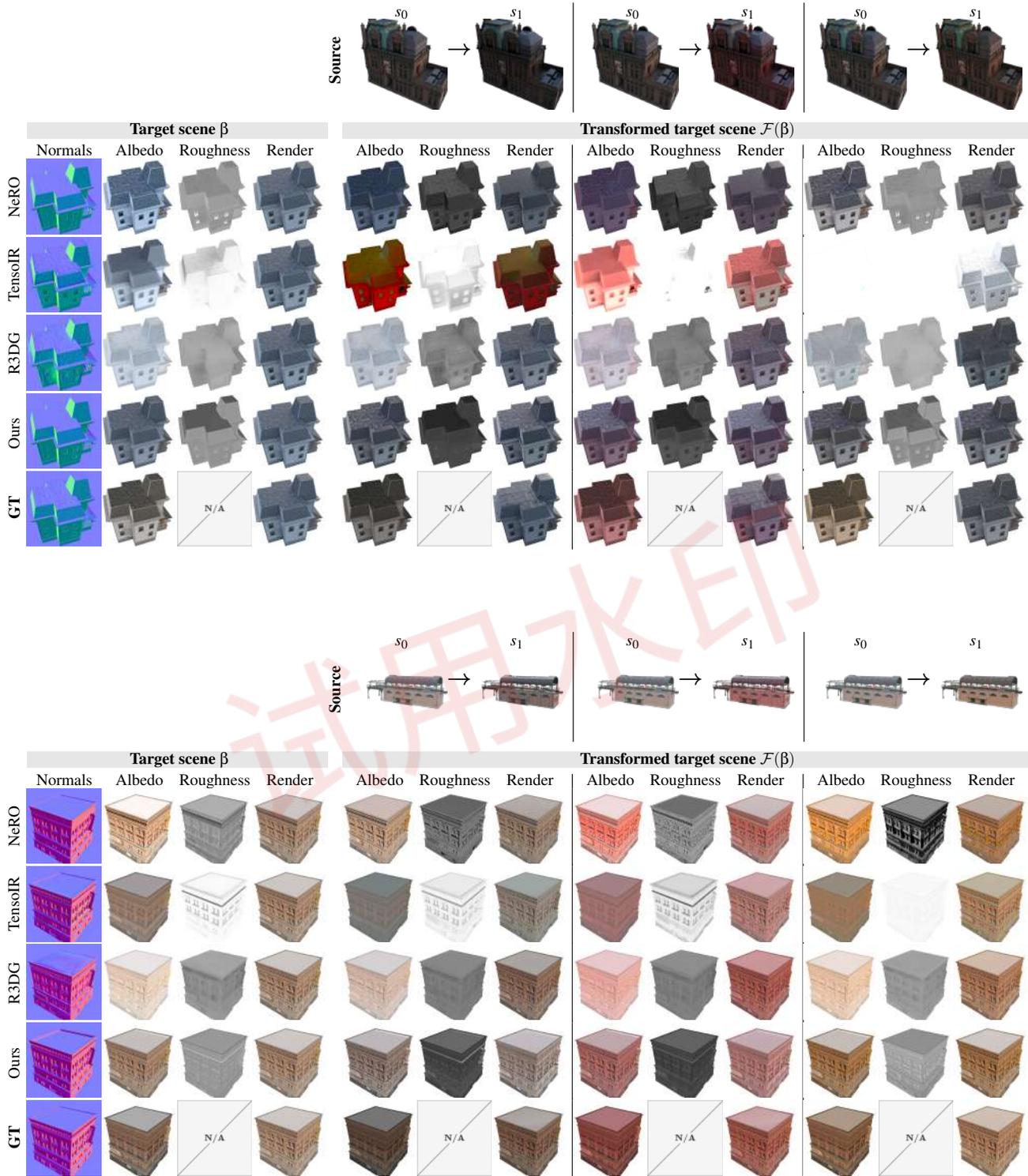


Figure 15: Qualitative material transforms results. We show qualitative results when synthesizing novel views with the learned transform function \mathcal{F} . For each sub-figure, we show in the top row the observed transform on the source scenes (s_0, s_1), with three possible transformations: T_1 , T_2 , and T_3 column-wise. On the left, we show the optimization results of the target scene, and on the right, the transformed BRDF below the corresponding three source transforms.

Acknowledgments. This work was funded by the French Agence Nationale de la Recherche (ANR) with the project SIGHT (ANR-20-CE23-0016) and performed with HPC resources from GENCI-IDRIS (Grant AD011014389R1). We thank Haian Jin for the useful discussion, and Mohammad Fahes and Anh-Quan Cao for their helpful comments.

References

- [BBJ*21] BOSS M., BRAUN R., JAMPANI V., BARRON J. T., LIU C., LENSCH H. P.: Nerf: Neural reflectance decomposition from image collections. In *Int. Conf. Comput. Vis.* (2021). 2
- [BHE23] BROOKS T., HOLYNSKI A., EFROS A. A.: Instructpix2pix: Learning to follow image editing instructions. 6
- [CT82] COOK R. L., TORRANCE K. E.: A reflectance model for computer graphics. In *ACM Trans. Graph.* (1982). 4
- [CXG*22] CHEN A., XU Z., GEIGER A., YU J., SU H.: Tensorf: Tensorial radiance fields. In *Eur. Conf. Comput. Vis.* (2022). 3
- [FLNP*24] FISCHER M., LI Z., NGUYEN-PHUOC T., BOZIC A., DONG Z., MARSHALL C., RITSCHER T.: Nerf analogies: Example-based visual attribute transfer for nerfs. In *IEEE Conf. Comput. Vis. Pattern Recog.* (2024). 2
- [FSV*23] FAN Y., SKOROKHODOV I., VOYNOV O., IGNATYEV S., BURNAEV E., WONKA P., WANG Y.: Factored-neus: Reconstructing surfaces, illumination, and materials of possibly glossy objects. 2023. 2
- [Gat20] GATIS D.: Rembg: a tool to remove images background. <https://github.com/danielgatis/rembg>, 2020. 5
- [GGL*23] GAO J., GU C., LIN Y., ZHU H., CAO X., ZHANG L., YAO Y.: Relightable 3d gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing. *arXiv:2311.16043* (2023). 5
- [GTR*06] GU J., TU C.-I., RAMAMOORTHY R., BELHUMEUR P., MATUSIK W., NAYAR S.: Time-varying surface appearance: Acquisition, modeling and rendering. In *ACM Trans. Graph.* (2006). 2
- [HHM22] HASSELGREN J., HOFMANN N., MUNKBERG J.: Shape, light, and material decomposition from images using monte carlo rendering and denoising. 2
- [HTE*23] HAQUE A., TANCİK M., EFROS A., HOLYNSKI A., KANAZAWA A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Int. Conf. Comput. Vis.* (2023). 2
- [JLX*23] JIN H., LIU I., XU P., ZHANG X., HAN S., BI S., ZHOU X., XU Z., SU H.: Tensorf: Tensorial inverse rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.* (2023). 1, 2, 3, 4, 5, 7
- [Kar13] KARIS B.: Real shading in unreal engine 4 by. 4
- [LCL*23] LIANG R., CHEN H., LI C., CHEN F., PANNEER S., VIJAYKUMAR N.: Envidr: Implicit differentiable renderer with neural environment lighting. In *Int. Conf. Comput. Vis.* (2023). 2
- [LLF*23] LI Y., LIN Z.-H., FORSYTH D., HUANG J.-B., WANG S.: Climatenerf: Extreme weather synthesis in neural radiance field. In *Int. Conf. Comput. Vis.* (2023). 2
- [LWL*23] LIU Y., WANG P., LIN C., LONG X., WANG J., LIU L., KOMURA T., WANG W.: Nero: Neural geometry and brdf reconstruction of reflective objects from multiview images. In *ACM Trans. Graph.* (2023). 2, 3, 4, 5
- [LWZW24] LI J., WANG L., ZHANG L., WANG B.: Tensosdf: Roughness-aware tensorial representation for robust geometry and material reconstruction, 2024. 2
- [LZC*23] LIU K., ZHAN F., CHEN Y., ZHANG J., YU Y., SADDİK A. E., LU S., XING E.: Stylerf: Zero-shot 3d style transfer of neural radiance fields. In *IEEE Conf. Comput. Vis. Pattern Recog.* (2023). 2
- [MAT*24] MA L., AGRAWAL V., TURKI H., KIM C., GAO C., SANDER P., ZOLLHÖFER M., RICHARDT C.: Specnerf: Gaussian directional encoding for specular reflections. 2
- [MHS*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 *IEEE Conf. Comput. Vis. Pattern Recog.* (2022). 2
- [MPBM03] MATUSIK W., PFISTER H., BRAND M., MCMILLAN L.: A data-driven reflectance model. *ACM Trans. Graph.* (2003). 2
- [MST*20] MILDENHALL B., SRINIVASAN P. P., TANCİK M., BARRON J. T., RAMAMOORTHY R., NG R.: Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. 2, 4
- [NSO24] NARUMOTO T., SANTO H., OKURA F.: Synthesizing time-varying brdfs via latent space. In *Eur. Conf. Comput. Vis.* (2024). 8
- [RSKS24] RADL L., STEINER M., KURZ A., STEINBERGER M.: LAENeRF: Local Appearance Editing for Neural Radiance Fields. In *IEEE Conf. Comput. Vis. Pattern Recog.* (2024). 2
- [SDZ*21] SRINIVASAN P. P., DENG B., ZHANG X., TANCİK M., MILDENHALL B., BARRON J. T.: Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.* (2021). 2
- [SF16] SCHÖNBERGER J. L., FRAHM J.-M.: Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2016). 5
- [SSR*07] SUN B., SUNKAVALLI K., RAMAMOORTHY R., BELHUMEUR P. N., NAYAR S. K.: Time-varying brdfs. In *IEEE Trans. Vis. Comput. Graph.* (2007). 2
- [SZPF16] SCHÖNBERGER J. L., ZHENG E., POLLEFEYS M., FRAHM J.-M.: Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)* (2016). 5
- [TMS*23] TOSCHI M., MATTEO R. D., SPEZIALETTI R., GREGORIO D. D., STEFANO L. D., SALTI S.: Relight my nerf: A dataset for novel view synthesis and relighting of real world objects. In *IEEE Conf. Comput. Vis. Pattern Recog.* (2023). 2, 5
- [UAS*24] UMMENHOFER B., AGRAWAL S., SEPULVEDA R., LAO Y., ZHANG K., CHENG T., RICHTER S., WANG S., ROS G.: Objects with lighting: A real-world dataset for evaluating reconstruction and rendering for object relighting. In *3DV* (2024). 2
- [VHM*22] 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 *IEEE Conf. Comput. Vis. Pattern Recog.* (2022). 2, 4
- [WHZL24] WANG H., HU W., ZHU L., LAU R. W. H.: Inverse rendering of glossy objects via the neural plenoptic function and radiance fields. In *IEEE Conf. Comput. Vis. Pattern Recog.* (2024). 2
- [YZL*22] YAO Y., ZHANG J., LIU J., QU Y., FANG T., MCKINNON D., TSIN Y., QUAN L.: Neilf: Neural incident light field for physically-based material estimation. In *Eur. Conf. Comput. Vis.* (2022). 2
- [ZLW*21] ZHANG K., LUAN F., WANG Q., BALA K., SNAVELY N.: Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *IEEE Conf. Comput. Vis. Pattern Recog.* (2021). 2
- [ZSD*21] ZHANG X., SRINIVASAN P. P., DENG B., DEBEVEC P., FREEMAN W. T., BARRON J. T.: Nerfactor: neural factorization of shape and reflectance under an unknown illumination. In *ACM Trans. Graph.* (2021). 2, 5
- [ZSH*22] ZHANG Y., SUN J., HE X., FU H., JIA R., ZHOU X.: Modeling indirect illumination for inverse rendering, 2022. 2
- [ZXY*23] ZHANG Y., XU T., YU J., YE Y., WANG J., JING Y., YU J., YANG W.: Nemf: Inverse volume rendering with neural microflake field. In *Int. Conf. Comput. Vis.* (2023). 2
- [ZYL*23] ZHANG J., YAO Y., LI S., LIU J., FANG T., MCKINNON D., TSIN Y., QUAN L.: Neilf++: Inter-reflectable light fields for geometry and material estimation. In *Int. Conf. Comput. Vis.* (2023). 2
- [ZZW*24] ZHUANG Y., ZHANG Q., WANG X., ZHU H., FENG Y., LI X., SHAN Y., CAO X.: Neai: A pre-convoluted representation for plug-and-play neural ambient illumination. In *AAAI* (2024). 2