DOI: 10.1111/cgf.14765
EUROGRAPHICS 2023 / K. Myszkowski and M. Nießner
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 2

In-the-wild Material Appearance Editing using Perceptual Attributes

J. Daniel Subias[1] and M. Lagunas[2]

[1]Universidad de Zaragoza, I3A, Spain
[2]Amazon, Spain

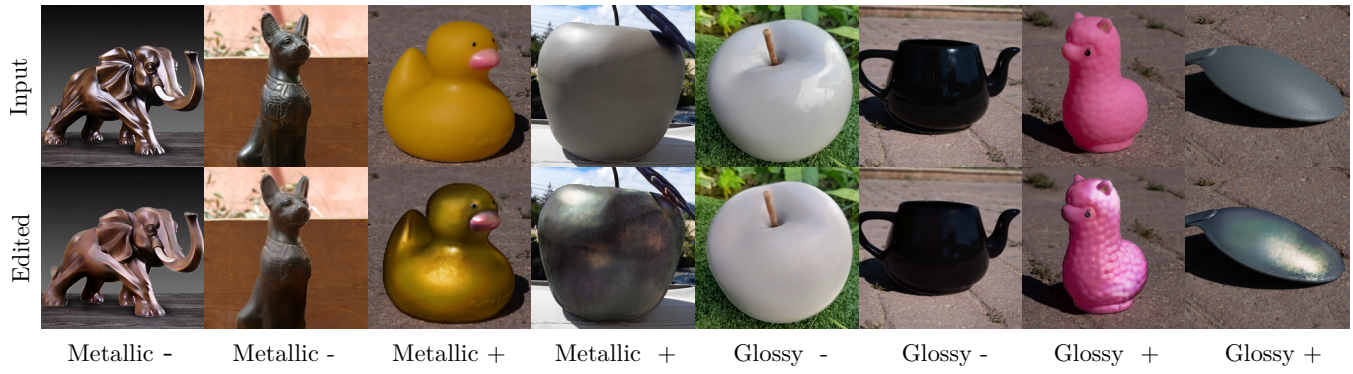

Figure 1: *Our approach enables intuitive appearance editing of high-level perceptual attributes. Our framework takes as an input a single image of an object (top) and produces high-quality edits of material attributes such as glossy or metallic, while preserving the geometrical structure and details (bottom). The"+" and "-" indicate whether the target high-level perceptual attribute is increased or decreased.*

Abstract

Intuitively editing the appearance of materials from a single image is a challenging task given the complexity of the interactions between light and matter, and the ambivalence of human perception. This problem has been traditionally addressed by estimating additional factors of the scene like geometry or illumination, thus solving an inverse rendering problem and subduing the final quality of the results to the quality of these estimations. We present a single-image appearance editing framework that allows us to intuitively modify the material appearance of an object by increasing or decreasing high-level perceptual attributes describing such appearance (e.g., glossy or metallic). Our framework takes as input an in-the-wild image of a single object, where geometry, material, and illumination are not controlled, and inverse rendering is not required. We rely on generative models and devise a novel architecture with Selective Transfer Unit (STU) cells that allow to preserve the high-frequency details from the input image in the edited one. To train our framework we leverage a dataset with pairs of synthetic images rendered with physically-based algorithms, and the corresponding crowd-sourced ratings of high-level perceptual attributes. We show that our material editing framework outperforms the state of the art, and showcase its applicability on synthetic images, in-the-wild real-world photographs, and video sequences.

CCS Concepts

• ***Computing methodologies*** → ***Machine learning; Computer graphics; Image manipulation;***

1. Introduction

Humans are visual creatures, visual data play a key role in the way we perceive and understand the world around us. We are able of recognizing materials, understanding their appearance, and reasoning about other physical properties effortlessly, just briefly looking at them. The visual appearance of a material — whether it appears *glossy*, *metallic*, or *rough* — is one of the key properties that determine how we manipulate and interact with objects. However, such visual appearance is formed by a complex multidimensional interaction involving the material properties themselves, but also other

confounding factors like geometry or illumination, that can affect our perception. Unfortunately, such underlying perceptual process is not yet completely understood [And11, Fle14, MB10].

Given the large amount of dimensions that our perception involves, many works have focused on editing material appearance by estimating the known parameters of the scene (material, geometry, and illumination), thus solving an inverse rendering problem where the material is later modified [BBP21, YX16]. However, this approach faces several problems: noise in the estimation of scene parameters can yield uncanny editing results. Moreover, the user should understand the intricacies of the material formulation to be able to edit. Techniques and hardware to acquire material appearance are gaining accuracy, speed, and efficiency [NJR15, AWL15, DJ18]; creating a data-driven shift for appearance editing techniques [SGM*16, ZZW*21, ZFWW20] . These methods still present similar problems as using inverse rendering. There exists a disconnection between the mathematical formulation of material appearance and human-friendly parameters. The captured data is machine friendly but not human friendly. Editing material appearance given a single in-the-wild photograph, and using intuitive perceptual attributes is therefore a challenging task that remains to be solved.

In this work, we present an image-based framework that does not rely on any physically-based rendering but instead modifies directly the material appearance in the image space. It takes a certain image of an object as input and modifies the appearance based on varying the value of the desired high-level perceptual attribute describing appearance (see Figure 1). Recently, the popularization of generative deep learning models had allow us to design data-driven frameworks for image editing [LDX*19, HZK*19, LZU*17]. Since the image cues that drive the perception of such attributes can not be captured in a few image statistics, we rely on such generative neural networks to learn their relationship with material appearance and generate novel edited images [SAF21, LSGM21].

We devise a generative architecture that allows to intuitively edit material appearance just from a single, in-the-wild, image and given the value of the target perceptual attribute that we want to manipulate. To train our framework, a first approach might be collecting pairs of images the originals and the edited ones, where the edited examples were produced given a target high-level perceptual attribute value. This approach is not only tedious but also hinders the training process. We follow the work from Delanoy et al. [DLC*22], which is based on training a two-step generative framework using a wide dataset composed of a large variety of images, paired with high-level perceptual ratings obtained through user studies. However, this method needs additional geometry information (as a normal map) of the target object. We thus devise an encoder-decoder architecture that allows us to send the relevant high-frequency information of the object's shape from the encoder to the decoder thanks to the STU cell [LDX*19] and a novel loss function. Thus, removing the need for the normal map as the input and not subduing our results in in-the-wild photographs to having good estimations of the normal map.

We trained two version of our framework focusing on two high-level perceptual attributes that are both common and easy to understand: *metallic* and *glossy*. We evaluate the consistency of our edits on a wide variety of scenes with different illuminations, materials, and geometries; and compare our results with the work from Delanoy et al. [DLC*22]. We observe that our method, despite resorting to simpler input and having a simpler architecture, obtains more realistic material appearance edits of the input image. We also assess the temporal consistency by editing video sequences composed of frames rendered using unseen illumination, geometry, and materials. We observe that our framework is capable of producing coherent outputs even when the additional temporal dimension is included.

2. Related Work

2.1. Visual Perception

Understanding how our visual system interprets our world is a longstanding goal in fields like computer graphics or applied optics [NOGRR21], that is yet to be understood [FDA01, FDA03]. Our visual perception of an object is guided by its material properties; but, also involves factors such as geometry [VLD07, HFM16], light conditions [HLM06, KFB10, CK15] or motion [DFY*11, MLMG19]. To reduce the dimensionality of the problem, previous work have focused on understanding single, high-level appearance properties like *glossiness* [PFG00, WAKB09, CK15], translucency [GXZ*13, XZG*20, GWA*15] and softness [SFV20, CDD21]; or draw inspiration from artists' implicit understanding of the key visual cues that guide visual perception [DCWP19, DSMG21]. Recent works suggest that material perception may be driven by complex non-linear statistics better approximated by highly non-linear models such as neural networks [FS19, SAF21, DLG*20, LMS*19]. Inspired by this, we propose a deep-learning-based framework for material appearance editing that relies on images paired with human judgments about high-level perceptual attributes to be trained.

2.2. BRDF-based Material Editing

Editing material appearance is a complex task since there is a disconnection between our perception and materials' physical properties [FWG13, TFCRS11, CK15]. Non-parametric models such as SVBRDFs are hard to edit. Different approaches have been proposed, such as inverse shading trees [LBAD*06], procedural models [HHD*22], or using deep-learning techniques [DDB20]. Several perceptually-based frameworks have been proposed to provide intuitive controls over parametric [FPG01, PFG00] and non-parametric appearance models [SGM*16, MGZ*17, HGC*20]. Unfortunately, these models only capture material properties, while leaving out other scene parameters that also drive our perception of material appearances, such as geometry and illumination [LSGM21]. Another line of work proposes an intrinsic decomposition of the scene to manipulate materials [BM15, YS19]. More recently, NeRF [MST*20] based approaches for such decomposition are allowing for unprecedented levels of realism [BBJ*21, ZSD*21, SDZ*21]. However, these methods only provide a new material definition that can later be used in a specific 3D scene but do not allow to edit it. Our work presents an intuitive framework that directly uses in-the-wild images and is capable of

editing the appearance of materials using high-level attributes such as *glossiness*.

2.3. Image-based Material Editing

Image-based material editing attempts to manipulate the pixel values directly into the image space. Several interactive frameworks have been proposed to provide users more intuitive controls over the target editing regions of the image, selecting them just from a few strokes [PL07, AP08, DTPG11]. The work from Khan et al. [KRFB06] proposes a single-image editing framework exploiting our tolerance to certain physical inaccuracies. Such approach was extended later to relight paintings [LMJH*10], consider global illumination and caustics [GSLM*08], or weathering effects [XWT*08]. More recent work, supported by the success of deep-learning-based methods, introduces editing approaches by factoring the image into shape, material, and reflectance [LCY*17, RRF*16, MLTFR19]. Splitting an image into separate components like frequency bands [BBPA15] or shading and reflectance [GMLMG12] has been a standard practice for image manipulation. Instead, we propose a generative-based editing framework without the need to decompose the input image, learning to edit directly the visual cues that drive our perception of high-level attributes. Generative Adversarial Networks (GANs) [GPAM*14] have been proposed to edit face attributes (i.e., hair color or gender) through the latent space [LZU*17, HZK*19, LDX*19]. Our framework works instead in the complex problem of material appearance, where confounding factors like geometry or illumination have a direct impact in our perception. Closer to ours, Delanoy et al. [DLC*22] introduce a generative framework for intuitive appearance editing using high-level perceptual attributes. However, their method requires two inputs: the image, and its normal map which yields non-photorealistic results for in in-the-wild images where the normal map is estimated. We devise a novel generative architecture that allows to keep the high-frequency information from the geometry of the input images, thus removing the need for the normal map as the input while obtaining superior performance.

3. Our Framework

This section describes our proposed framework for single-image appearance editing. We introduce our goal (Section 3.1), describe the mathematical model of the Selective Transfer Units (STU) cells (Section 3.2), and, explain the different modules that make our model architecture is built upon (Section 3.3).

3.1. Goal and Overview

Our goal is to generate an image $\mathbf{y}$ whose material appearance we want to edit from an input image $\mathbf{x}$ and a value $\mathbf{att}_t \in [0,1]$ of the target high-level perceptual attribute to edit (e.g., *glossy* or *metallic*). The target edited image $\mathbf{y}$ depicts the same object as $\mathbf{x}$ and elicits a visual appearance according to the value of the high-level perceptual attribute $\mathbf{att}_t$. For instance, more *glossy* if $\mathbf{att}_t$ is closer to 1 and less if closer to 0. To achieve it, we introduce a novel framework that relies on an encoder-decoder architecture G that encodes the image $\mathbf{x}$, and manipulates the latent space $\mathbf{z}$ together with the target attribute $\mathbf{att}_t$ to generate the edited image $\mathbf{y}$. A high-level overview of our framework is shown in Figure 2.

3.2. Selective Transfer Units

When using encoder-decoder architectures, it is common practice to send information between the encoder and decoder by using skip-connections [RFB15]. This allows us to keep high-frequency details in the generated image that are lost otherwise [KW13, HMP*17]. However, in image editing tasks that manipulate the latent space, adding skip connections hampers the editability of the model, where only the input image can be reconstructed [CCK*18, HZK*19, DLC*22]. To bridge this gap, we send information from the encoder to the decoder by selectively removing unnecessary data using Selective Transfer Units (STU) memory cells [LDX*19].

The STU architecture, illustrated in Figure 2, is a variant of the GRU [CvMG*14, CGCB14] and allows encoder-decoder architectures to keep the relevant information of the input image in the edited output when manipulating the latent space $\mathbf{z}$. Given the feature map of the l^{th} encoder layer denoted by $\mathbf{f}^l_{enc}$, the STU cell outputs an edited feature map $\mathbf{f}^l_t$, as is shown in Figure 2. Each STU cell receives information from the previous cell via a feature map $\hat{\mathbf{s}}^{l+1}$ (also called hidden state), which also contains information of the target attribute $\mathbf{att}_t$. The STU updates its internal hidden state denoted as $\mathbf{s}^l$ and sends this to the next STU cell. For further information about the mathematical formulation of STU cells refer to Appendix A.

3.3. Network Architecture

Our framework is a GAN-like model composed of a generator G and a discriminator D that is only used during training. Our goal is to leverage G to edit an in-the-wild input image $\mathbf{x}$ according to a target high-level perceptual attribute describing appearance $\mathbf{att}_t$ to generate an edited image $\mathbf{y}$ where

$$\mathbf{y} = G(\mathbf{x}, \mathbf{att}_t). \tag{1}$$

The generator G is composed of an encoder module G_{enc} that encodes the input image to a latent vector $\mathbf{z}$ and a decoder module G_{dec} that generates the edited image, both with the same number n of layers.

The encoder G_{enc} compresses the image $\mathbf{x}$ in a latent code $\mathbf{z}$, while storing the set of features maps $\mathbf{f} = \{\mathbf{f}^1_{enc}, \mathbf{f}^2_{enc}, ..., \mathbf{f}^{n-1}_{enc}\}$, generated by its convolutional layers. The latent code $\mathbf{z}$ corresponds to the feature map generated by the last convolutional layer such that $\mathbf{z} = \mathbf{f}^n_{enc}$. The decoder G_{dec} reconstructs the edited image $\mathbf{y}$ from the latent code $\mathbf{z}$ concatenated with the target high-level perceptual attribute $\mathbf{att}_t$. To keep the high frequencies from the input in the edited image $\mathbf{y}$, we send the feature maps $\mathbf{f}$ via skip-connections. However, performing this process without additionally processing the information in the features affects the editing ability of the generator G. Thus, we introduce an STU cell in each skip-connection where G_{st} denotes the set of STU cells of the framework, and G^l_{st} corresponds to the STU cell of the l^{th} layer. The STU cells edit the set of feature maps $\mathbf{f}$ from the input, generating a set of edited ones

$\mathbf{f}_t = \{\mathbf{f}_t^1, \mathbf{f}_t^2, ..., \mathbf{f}_t^{n-1}\}$. Since the STU cell of the deepest skip connection G_{st}^{n-1} does not receive a hidden state $\hat{\mathbf{s}}$ from another STU cell, G_{st}^{n-1} takes as input the latent code $\mathbf{z}$ concatenated with the target attribute $\mathbf{att}_t$ to edit the feature map $\mathbf{f}_{enc}^{n-1}$, as is shown in Figure 2. For further details of our architecture refer to Appendix B.

4. Learning to Edit Material Appearance

Training GAN-like models is a complex task that requires careful tuning of the hyperparameters. We first describe the dataset of images, with paired crowd-sourced ratings of high-level perceptual attributes, that we used to train our framework (Section 4.1). Then we introduce the loss function that allows us to faithfully edit material appearance according to the target perceptual attribute (e.g., *glossy*) (Section 4.2), and last, we describe the technical details of the training process (Section 4.3).

4.1. Training Dataset

Generating ground-truth data from analytical BRDF models is not a robust approach to training our framework, since their parameters are not aligned with our perception [NDM06, WAKB09]. In addition, our framework would learn to edit based on variations in a physical parameter (e.g., roughness) and not on variations in our perception. Thus, we leverage the dataset of Delanoy et. al [DLC*22], designed for material appearance perception tasks. This dataset based on the Lagunas et al. dataset [LMS*19] contains renderings of 13 different geometries, illuminated by 7 captured real-world illuminations [Deb]. Renders have been made with the physically-based renderer Mitsuba [Jak10] using 100 different BRDFs from the Merl dataset [MPBM03]. For each combination of material $\times$ shape $\times$ illumination, 5 different images with slight variations in the viewpoint (randomly sampled within a 45 degrees cone around the original viewpoint) have been rendered, as is shown in Figure 3. The dataset has 45,500 images (13 geometries $\times$ 100 materials $\times$ 7 illuminations $\times$ 5 views).

Each scene is described by a set of high-level perceptual attributes (i.e., *plastic*, *rubber*, *metallic*, *glossy*, *bright*, *rough*, and the strength and sharpness of *reflections*) rated on a normalized Likert scale, from 0 to 1. A total of 2,600 paid subjects participated in the study, each of them rating 15 different random images. The final dataset gathers a perceptual rating per viewpoint. To reduce the presence of noise and outliers, we use the median value pooled over the 5 viewpoints and the shape. Then the attribute values are different for each material and illumination.

Data Augmentation To have more diversity in the images and help our model generalize better, we add a data augmentation pipeline. First, we resize the images to 512×512 px; then we flip and rotate them 90 degrees randomly, and we finally crop them to 480×480 px. In addition, to reduce the bias on the colors BRDF we represent images on the HSV color space and make a shift in the Hue and Saturation channels. Finally, the input is resized to a size of 256×256 px. to remove the background. Within each input image, the background is removed by applying a mask of the object's silhouette. Each image in the training dataset is paired with a high-level perceptual attribute $\mathbf{att}_s \in [0,1]$; however, with our framework, we want to edit the input images with a different target attribute $\mathbf{att}_t$. During training, we sample $\mathbf{att}_t$ randomly using a uniform distribution $\mathbf{U}([0,1])$ allowing the generator G to produce edited images different from the input to trick the discriminator D.

4.2. Loss Functions

We adopt the adversarial training proposed by He et al. [HZK*19] and introduce a GAN model where the discriminator D has two branches D_{adv} and D_{att}. D_{adv} consists of five convolution layers to predict whether an image is fake (edited) or real. D_{att} shares the convolution layers with D_{adv} and, instead, predicts the high-level attribute value $\mathbf{att}_t$. Figure 4 shows a high-level scheme of our framework during training.

Adversarial Losses Since we do not know what are the ground-truth edited results, we use an adversarial loss [GPAM*14] aiming to generate edited results indistinguishable from real images. GANs are complex to train, partially due to the instability of the loss function proposed in the original formulation [ACB17]. Thus, we rely on the WGAN-GP [GAA*17] loss to alleviate such a problem. The discriminator D is trained to give a high score to real images and a low score to generated ones, aiming to disambiguate them, and tries to minimize the adversarial loss defined as:

$$\mathcal{L}_{D_{adv}} = \mathbb{E}_{\mathbf{x}}\left[D_{adv}(\mathbf{x})\right] - \mathbb{E}_{\mathbf{y}}\left[D_{adv}(\mathbf{y})\right] + \mathcal{L}_{GP}. \tag{2}$$

Formally, D should be a 1-Lipschitz continuous function. To keep this constraint, WGANs [ACB17] introduces a gradient penalty term defined as follows:

$$\mathcal{L}_{GP} = \lambda_1 \mathbb{E}_{\hat{\mathbf{x}}}\left[\left(||\nabla_{\hat{\mathbf{x}}} D_{adv}(\hat{\mathbf{x}})||_2 - 1\right)^2\right]. \tag{3}$$

Where λ_1 is the gradient penalty weight. The generator G is trained such that the discriminator D believes that generated images are real, giving them a high score. Therefore the generator's adversarial loss is given by:

$$\mathcal{L}_{G_{adv}} = \mathbb{E}_{\mathbf{y}}\left[D_{adv}(\mathbf{y})\right]. \tag{4}$$

Attribute Manipulation Losses Even though the ground truth is missing, the editing result has to elicit a visual impression according to the target perceptual attribute $\mathbf{att}_t$. Therefore, we introduce an attribute classifier D_{att} which learns to predict the attribute values $\mathbf{att}_s$ from the images that belong to the training dataset. Since we have to compare the distance from the predicted attributes by D_{att} to $\mathbf{att}_s$, the following attribute manipulation loss is computed and optimized by D_{att} during training:

$$\mathcal{L}_{D_{att}} = ||\mathbf{att}_s - D_{att}(\mathbf{x})||_1. \tag{5}$$

The generator G should produce images with similar looks to real ones to trick D, so its edits must be consistent with the target perceptual attributes $\mathbf{att}_t$, minimizing the following distance:

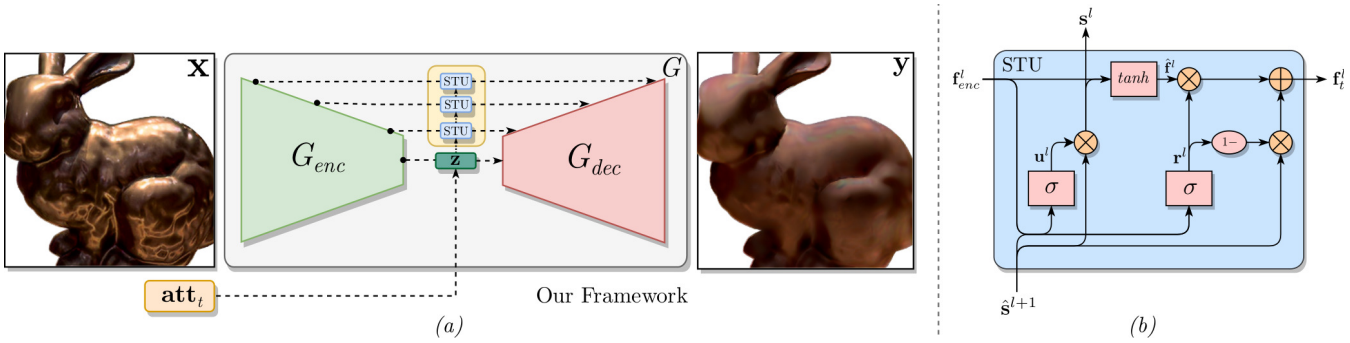

Figure 2: *(a) High-level overview of our framework. Our generator G is composed of an encoder G_{enc} and a decoder G_{dec}. It is capable of editing the input image* $\mathbf{x}$ *according to the target attribute* $\mathbf{att}_t$ *to generate the edited image* $\mathbf{y}$*. (b) The architecture of a single STU cell. As an input, it takes the feature map of the current layer* $\mathbf{f}^l_{enc}$ *and the hidden state of the previous cell* $\hat{\mathbf{s}}^{l+1}$*. It outputs the updated hidden state* $\mathbf{s}^l$ *and feature map* $\mathbf{f}^l_t$*.*

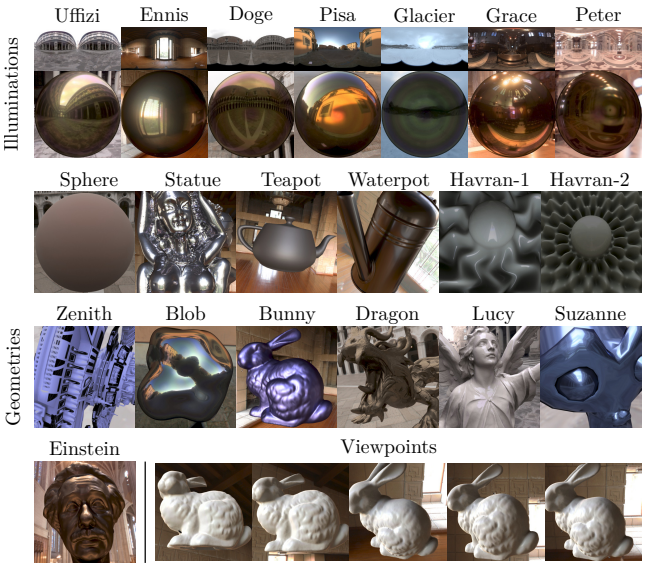

Figure 3: *Seven illuminations present in the training dataset and corresponding rendered spheres with the brass material (top). Thirteen scenes of the geometries present in the training dataset, and five viewpoints of the bunny geometry rendered using the nylon material and Ennis illumination (bottom).*

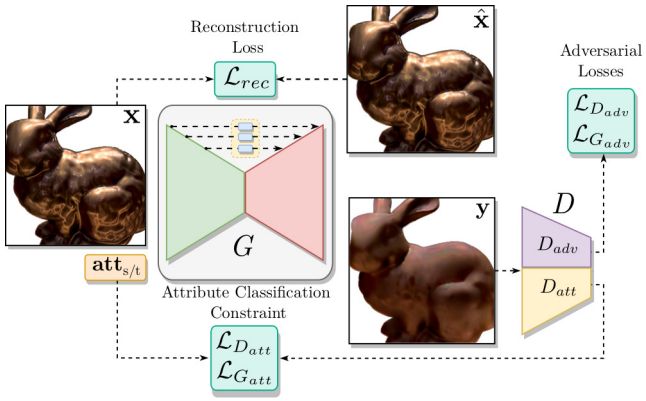

Figure 4: *Training scheme of our framework. The gray block represents our generator G, and the discriminator branches D_{adv} and D_{att} are illustrated by the purple and yellow blocks, respectively. Pointed arrows denote the parameters used as input for the training losses (Section 4.2).*

$$\mathcal{L}_{G_{att}} = ||\mathbf{att}_t - D_{att}(\mathbf{y})||_1. \tag{6}$$

Reconstruction Loss Each image in the training set has crowd-sourced ratings of high-level perceptual attributes $\mathbf{att}_s$, describing their appearance. Thus, if we ask our generator G to edit $\mathbf{x}$ according to the attribute $\mathbf{att}_s$, it should generate an edited image $\hat{\mathbf{x}}$ similar to the input. We minimize the following L1-norm during training:

$$\mathcal{L}_{rec} = ||\mathbf{x} - \hat{\mathbf{x}}||_1. \tag{7}$$

Final Loss Taking the above losses into account, the objectives to train the discriminator D and generator G can be formulated as:

$$\min_D \mathcal{L} = -\mathcal{L}_{D_{adv}} + \lambda_2 \mathcal{L}_{D_{att}}, \tag{8}$$

$$\min_G \mathcal{L} = -\mathcal{L}_{G_{adv}} + \lambda_3 \mathcal{L}_{G_{att}} + \lambda_4 \mathcal{L}_{rec}. \tag{9}$$

Where λ_2, λ_3, and λ_4 are the model trade-off parameters, that are tunned to yield values of the individual losses in a similar order of magnitude. Both G and D will try to minimize their objective function, however, once G has learned to trick D, the loss function of the latter will tend to increase until both models reach a stable equilibrium.

4.3. Training Details

We train the model using the ADAM optimizer [KB14] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The learning rate η is 2×10^{-4} for both G and D. The trade-off parameters are $\lambda_2 = 50$, $\lambda_3 = 100$ and $\lambda_4 = 1000$ for the Equations 8 and 9, and $\lambda_1 = 10$ for Equation 3 (see Appendix B for further details). At each training step, the discriminator iterates 7 times while the generator updates its parameters once. The memory module G_{st} is composed of 4 skip connections with STU cells between the encoder and decoder. All the experiments are computed using the PyTorch [PGM*19] library with cuDNN 7.1, running on an Nvidia GeForce RTX 3090 GPU. In total, training the whole framework takes two days.

5. Evaluation and Results

We evaluate our framework on a set of synthetic images and real photographs that have not been seen by the model during training. We start by describing the evaluation dataset(Section 5.1); validate the design choices of our framework with a series of ablation studies (Section 5.2); demonstrate the robustness of the proposed method by comparing the obtained results with varying geometry, illumination, or material (Section 5.3); and finally compare our method to the state of the art, obtaining better performance (Section 5.4).

5.1. Evaluation Dataset

To evaluate our framework we leverage the synthetic dataset used by Delanoy et al. [DLC*22], containing scenes with shapes and materials never seen during training. They have been rendered using eight shapes collected from online sources, four illuminations obtained from HDRI-Haven [Hdr], and eight materials coming from Dupuy and Jakob's database [DJ18]. Figure 5 shows a representative subset of the synthetic scenes. We also test our framework's performance with real-world photographs downloaded from online catalogs of decorative items and with in-the-wild mobile photos taken by us. We masked the object of interest using the online API Kalideo [Kal]. Figure 6 shows material appearance editing results using our framework sampling different values for the target attribute $\mathbf{att_t}$. We can see the consistency of our edits when the attribute varies for in-the-wild photographs.

5.2. Ablation Studies

We evaluate our design choices via a series of ablation studies. Our framework is based on a generative architecture that uses a discriminator D during training and STU cells [LDX*19] to obtain more realistic edited images while keeping the relevant high-frequency details of the input image. The model without the discriminator D or without STU cells are capable of learning to better reconstruct the input image. However, without D (w/o D) we are not able to properly edit the material appearance since no feedback about such edits exists. On the other hand, concatenating skip-connections directly, without including information from the attribute, hampers the decoder's ability to generate the edited image from the latent code $\mathbf{z}$ (w/o STUs). Figure 7, depicts how not using a discriminator D generates an image almost equal to the input, not allowing for editing. Besides, not including the STU cell does not convey the desired appearance according to the target high-level attribute in the final image.

Figure 5: *Synthetic dataset used to evaluate our framework. The images show eight geometries rendered with illumination and materials never seen in the training process.*

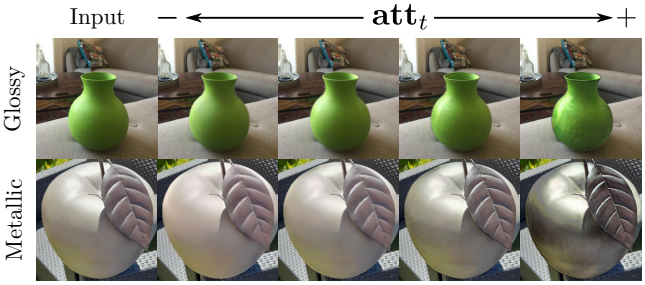

Figure 6: *Editing results by varying the metallic or glossy attributes. The "+" and "-" indicate whether the target high-level perceptual attribute is increased or decreased.*

5.3. Consistency of the Edits

We test the robustness of our framework exploring its editing ability through samples from the evaluation dataset. Since we have a wide collection of rendered images with diverse materials, illuminations, and geometries; we can fix a scene parameter (i.e., illumination, geometry, or material) and vary the other two. When working with in-the-wild photographs, scene parameters are unknown. We can not modify the material or geometry of the objects but we are able to change their illumination conditions by placing them elsewhere. Figure 8 (a), (b), (c) shows images edited by our framework for the *metallic* attribute while varying scene parameters for synthetic images. On the other hand, in (d) we vary the illumination conditions for in-the-wild photographs of the same object (for further results see the Supplementary Material). We can see how our framework produces consistent results in the different conditions, and with synthetic and in-the-wild photographs.

We also test the temporal consistency of our framework by editing two video sequences frame by frame for both *metallic* and *glossy* attributes.

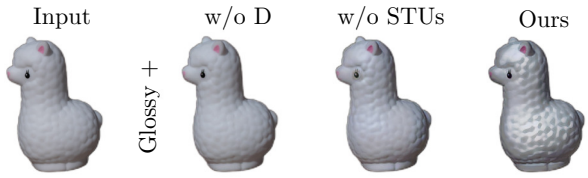

Figure 7: *Ablation studies where we analyze the impact of the different components of our framework. From left to right: input image, edited image without the discriminator D during training, edited image without using the STU cell, and our framework. We can see how by not using the discriminator D the framework is not able to edit and only reconstructs the input image, and using only skip connections, without the STU, hampers the ability to edit. The "+" indicates an increase of the target attribute.*

Table 1: *Average PSNR, SSIM, MSE and MAE reconstructing the input image on the synthetic dataset. We can see how our framework outperforms the method proposed by Delanoy et al. [DLC*22].*

Method	PSNR ↑	SSIM ↑	MSE ↓	MAE ↓
DLC*22	15.989	0.761	0.095	0.031
Ours	**27.388**	**0.967**	**0.002**	**0.023**

5.4. Comparison with the State of the Art

We compare our results against the method of Delanoy et al. [DLC*22]. We show results reconstructing the input image using the crowd-sourced perceptual attribute as the input, and editing the input using a different target perceptual attribute. Since the method of Delanoy et al. also needs the normal map as the input, we evaluate in-the-wild photographs with their estimated normal map (using Delanoy et al. estimator), and synthetic images with a perfectly normal map. Our method does not need the normal map as the input. We rely on qualitative and quantitative comparisons employing four metrics: Pixel-to-Signal Noise Ratio (PSNR), Structural Similarity Index (SSIM), Mean Squared Error (MSE), and Mean Absolute Error (MAE).

Quantitative Evaluation To evaluate the reconstruction ability we rely on the synthetic images with their crowd-sourced perceptual attributes. Table 1 shows we outperform the state of the art as a result of introducing STU cells in each skip-connection. As illustrated in Figure 9, our method keeps high-frequency details from the input image without the need of a normal map of the object's surface as the input. We can see that specular reflections are present on the reconstructed image while the previous method blurs them, keeping only low frequencies.

Image Editing In Figure 10 we can see a comparison between the edited images by our method and the one by Delanoy et al. [DLC*22] (for further comparisons see the Supplementary Material). Our approach learns to edit perceptual cues properly while objects' shape remains unchanged. The material appearance edits from Delanoy et al. [DLC*22] strongly depend on the shape of their estimated normal map [LAD*21]. This causes geometry details that are not present in the normal map not to be present in the edited image. Also, an inaccurate estimation of the normal map may deform the original shape, especially in in-the-wild photographs where geometries are usually highly complex (see Figure 11).

5.5. Comparison with Physically-based Rendering

We compare our results with physically-based rendering by varying the roughness parameter of the Principled BSDF [BS12, Bur15] in the interval $[0,1]$ and rendering different versions of the same scene (using geometry and illumination not present in our training dataset). The other parameters and the albedo constant and we rely on the physically-based renderer Mitsuba 3 [JSR*22] to generate the images. Then, with our method, we increase and decrease the gloss attribute using the rendered scene with a roughness value of 0.5 as the input. As we can see in Figure 12, our edits convey the overall appearance of the rendered images. However, we are less accurate when the gloss attribute is increased. This may be explained since removing existing information (highlights) is easier than introducing missing information without providing the environment map.

6. Limitations and Future Work

We have presented and validated a framework for intuitive material appearance editing using single, in-the-wild images. We relied on a large set of images paired with crowd-sourced ratings of high-level perceptual attributes to train our framework. We use a generative neural network and devise a loss function that allows us to learn how to edit material appearance based on such high-level attributes, without any pairs of original and edited images. Our results show that the presented method can achieve realistic results, almost on par with real photographs, on a wide variety of different inputs. However, our method is not free of limitations. As we can see in Figure 13, when using input photographs that feature highly specular highlights, while able to convey the appearance of the target high-level perceptual attribute, our framework may struggle to edit them. Instead of considering the original albedo to perform edit the highlights when *glossiness* is decreased, the resulting edits feature a dimmed region.

Material appearance perception and editing pose many challenges that are not fully investigated in this work. We propose a data-driven approach for intuitive material editing where we have relied on an existing dataset that was rendered using MERL [MPBM03] material measurements. MERL just contains 100 real-world isotropic BRDFs, we have tried to increase the variety of such data using different data augmentation strategies during training time. This has allowed us to obtain plausible results, nevertheless, exploring more complex material representations, or including other newly measured material datasets [DJ18, FV14, SCW*21] could help obtain more universal models for material editing. Also, the high-level perceptual ratings in the dataset come from online surveys. While we may identify *metallic* and *glossy* as different properties, there is a certain degree of correlation in user answers between those attributes. In Figure 14 we show a real photograph edited by our framework. There, we observe that while yielding different results, the increased *glossy* and

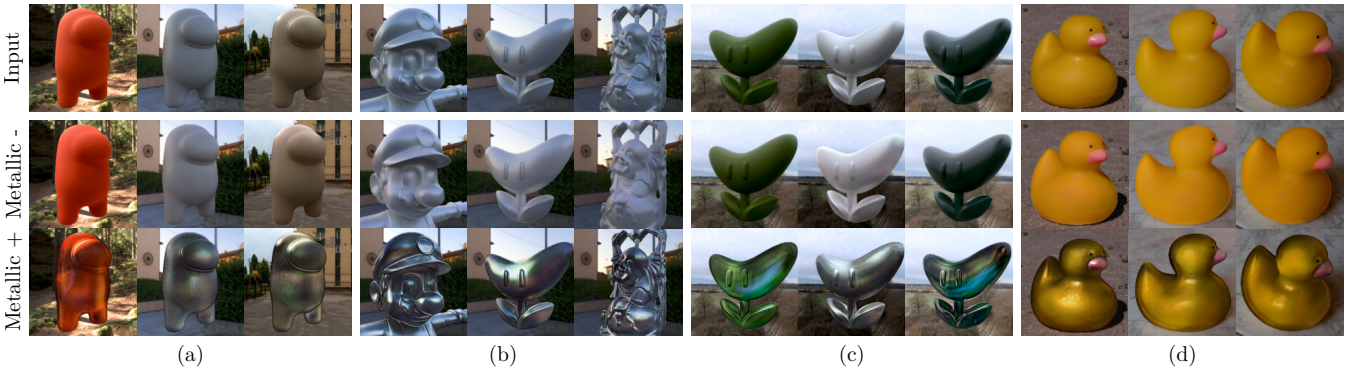

Figure 8: *Examples of edited images by our framework. (a) Renders of the Among Us geometry under three different illuminations, and two constant materials. (b) Renders of Mario, Boomerang Flower, and Buddha geometries with the same material and light conditions. (c) Renders of the Boomerang Flower geometry with the same illumination but different materials. (d) Photographs of a rubber duck taken in different places under three light conditions. Our framework is capable of producing compelling and consistent edits in all cases. The "+" corresponds to an increase in the target metallic attribute value* $\mathbf{att}_t$*, while the "-" corresponds to a decrease.*

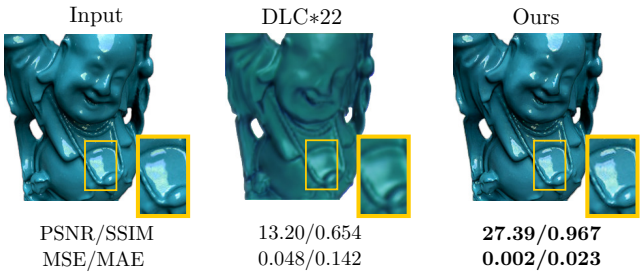

Figure 9: *A demonstration of the reconstruction quality for Buddha geometry. Yellow insets show regions of the object's surface with specular reflections.*

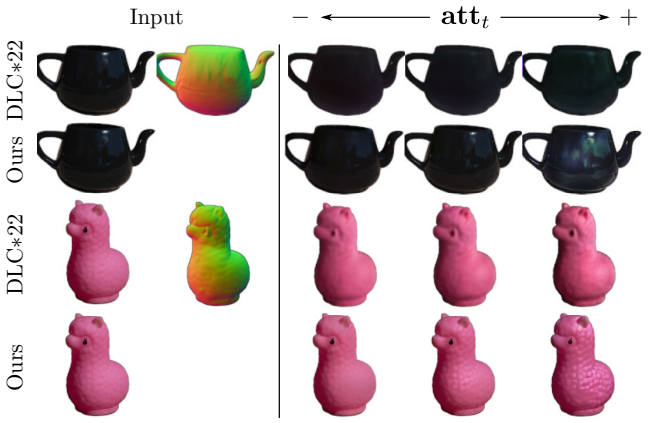

Figure 10: *Comparison editing the glossy attribute using the method of Delanoy et al. [DLC*22] and our framework for two in-the-wild photographs. Our framework only requires the photograph as the input while the work of Delanoy et al. needs to estimate the normal map. We can see how our method better recovers the glossy appearance of the object when edited. Besides, it is able to recover better high-frequency details. The "+" and "-" indicate whether the target high-level perceptual attribute increased or decreased.*

metallic edits share a certain resemblance. Investigating other attributes, or different perceptual data collection strategies may yield improved performance and more intuitive tools. We also observe that the collected perceptual ratings are not distributed uniformly. This result may come from both, a bias in the dataset, and humans' perceptual behavior. As a result, our framework may not learn to uniformly edit material appearance when varying the target attribute (see Supplementary Material). Another potential approach to generate the dataset would be to rely on a BRDF model to generate images, and input a parameter of this material model (e.g., roughness) as the target attribute. However, while the framework may be trained using this data, in this work we are addressing the more complex problem of editing appearance from high-level perceptual attributes where the users have factored in all potential confounding factors of perception (including the material model) in their ratings [LSGM21]. Last, the generative neural network used in this work has been trained on the limit of our hardware. To use higher-resolution images, one could use the proposed method (and its weights) as a backbone, add additional layers, and fine-tune the model while increasing the resolution size of the input. This could be repeated in an iterative process until we get the desired resolution size [KALL17]. However, although this is a possible approach, its effectiveness requires further investigation. We have created a framework that is trained for each attribute. Developing a novel methodology that would allow to manipulate an *appearance vector* can help to have a more comprehensive description of material appearance, and more intuitive edits. In addition to the results we have shown, we hope that our work can inspire additional research and different applications around material appearance. We will make our code available for further experimentation, to facilitate the exploration of these possibilities.

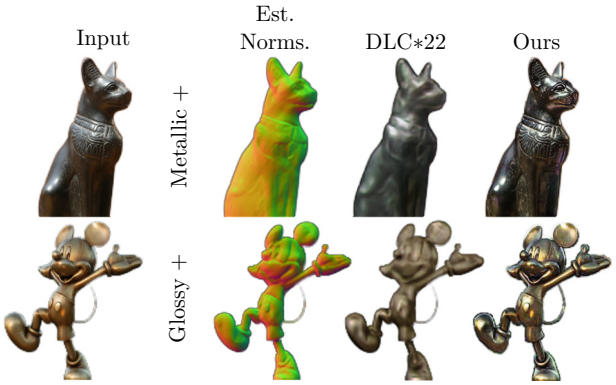

Figure 11: *Delanoy et al. [DLC*22] output is highly dependent to the estimated normal map of the input image. A low-frequency estimation of the normal map yields dull edits of material appearance. The "+" indicates an increase of the target attribute.*

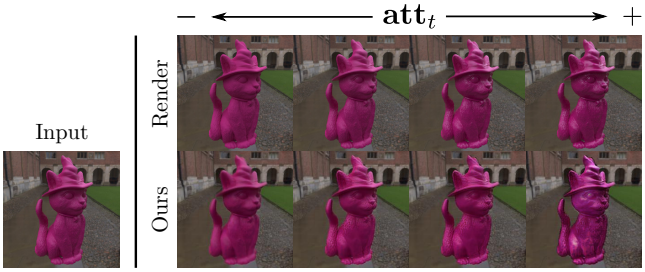

Figure 12: *Results varying the roughness parameter of the Principled BSDF [BS12, Bur15] (top row) and using our framework (bottom row) to edit the gloss high-level perceptual attribute. The "+" and "-" indicate whether the roughness parameter and the target high-level perceptual attribute increased or decreased.*

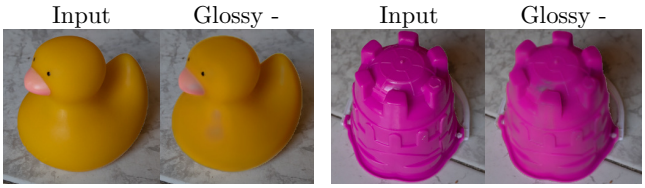

Figure 13: *Example of two failure cases of our framework for the glossy attribute on photographs. The "-" indicates a value of the target attribute set to 0.*

Figure 14: *Comparison between the metallic and glossy attribute. The "+" indicates that* **att**$_t$ *increases, while "-" indicates that it decreases. It is interesting to observe that if the metallic attribute is decreased, the gloss is not removed from the image.*

7. Acknowledgements

This project has received funding from the Government of Aragon's Departamento de Ciencia, Universidad y Sociedad del Conocimiento through the Reference Research Group "Graphics and Imaging Lab", the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 956585 (PRIME), the CHAMELEON project (European Union's Horizon 2020, European Research Council, grant agreement No. 682080), and MCIN/AEI 10.13039/501100011033 through Project PID2019-105004GB-I00. We want to thank the Institute for Pure and Applied Mathematics, Universidad Politecnica de Valencia, for the provided computational resources. Also, we would like to thank Daniel Martin and Ana Serrano for proofreading, and Adolfo Munoz, Julio Marco, and Diego Gutierrez for the discussions about the project. Manuel Lagunas' work was done as part of a collaboration with Universidad de Zaragoza and it does not interfere with his position at Amazon.

References

[ACB17] ARJOVSKY M., CHINTALA S., BOTTOU L.: Wasserstein generative adversarial networks. In *Proc. International Conference on Machine Learning (ICML)* (06–11 Aug 2017), Precup D., Teh Y. W., (Eds.), vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 214–223. 4

[And11] ANDERSON B. L.: Visual perception of materials and surfaces. *Current Biology 21*, 24 (2011), R978–R983. 2

[AP08] AN X., PELLACINI F.: Appprop: All-pairs appearance-space edit propagation. In *ACM SIGGRAPH 2008 Papers* (New York, NY, USA, 2008), SIGGRAPH '08, Association for Computing Machinery. 3

[AWL15] AITTALA M., WEYRICH T., LEHTINEN J.: Two-shot svbrdf capture for stationary materials. *ACM Trans. on Graphics 34*, 4 (jul 2015). 2

[BBJ*21] BOSS M., BRAUN R., JAMPANI V., BARRON J. T., LIU C., LENSCH H. P.: Nerd: Neural reflectance decomposition from image collections. In *Proc. International Conference on Computer Vision (ICCV)* (October 2021), pp. 12684–12694. 2

[BBP21] BATI M., BARLA P., PACANOWSKI R.: An inverse method for the exploration of layered material appearance. *ACM Trans. on Graphics 40*, 4 (jul 2021). 2

[BBPA15] BOYADZHIEV I., BALA K., PARIS S., ADELSON E.: Band-sifting decomposition for image-based material editing. *ACM Trans. on Graphics 34*, 5 (nov 2015). 3

[BM15] BARRON J. T., MALIK J.: Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence 37*, 8 (2015), 1670–1687. 2

[BS12] BURLEY B., STUDIOS W. D. A.: Physically-based shading at disney. In *ACM SIGGRAPH* (2012), vol. 2012, vol. 2012, pp. 1–7. 7, 9

[Bur15] BURLEY B.: Extending the disney brdf to a bsdf with integrated subsurface scattering. *SIGGRAPH Course: Physically Based Shading in Theory and Practice. ACM, New York, NY 19* (2015). 7, 9

[CCK*18] CHOI Y., CHOI M., KIM M., HA J.-W., KIM S., CHOO J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proc. Computer Vision and Pattern Recognition (CVPR)* (June 2018). 3

[CDD21] CAVDAN M., DREWING K., DOERSCHNER K.: Materials in action: The look and feel of soft. *bioRxiv* (2021). 2

[CGCB14] CHUNG J., GÜLÇEHRE Ç., CHO K., BENGIO Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR abs/1412.3555* (2014). 3, 12

[CK15] CHADWICK A., KENTRIDGE R.: The perception of gloss: A review. *Vision Research 109* (2015), 221–235. Perception of Material Properties (Part I). 2

[CvMG*14] CHO K., VAN MERRIËNBOER B., GULCEHRE C., BAHDANAU D., BOUGARES F., SCHWENK H., BENGIO Y.: Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proc. Empirical Methods in Natural Language Processing (EMNLP)* (Doha, Qatar, Oct. 2014), Association for Computational Linguistics, pp. 1724–1734. 3, 12

[DCWP19] DI CICCO F., WIJNTJES M. W. A., PONT S. C.: Understanding gloss perception through the lens of art: Combining perception, image analysis, and painting recipes of 17th century painted grapes. *Journal of Vision 19*, 3 (03 2019), 7–7. 2

[DDB20] DESCHAINTRE V., DRETTAKIS G., BOUSSEAU A.: Guided fine-tuning for large-scale material transfer. *Computer Graphics Forum 39*, 4 (2020), 91–105. 2

[Deb] DEBEVEC P.:. URL: https://vgl.ict.usc.edu/Data/HighResProbes/. 4

[DFY*11] DOERSCHNER K., FLEMING R., YILMAZ O., SCHRATER P., HARTUNG B., KERSTEN D.: Visual motion and the perception of surface material. *Current Biology 21*, 23 (2011), 2010–2016. 2

[DJ18] DUPUY J., JAKOB W.: An adaptive parameterization for efficient material acquisition and rendering. *ACM Trans. on Graphics 37*, 6 (dec 2018). 2, 6, 7

[DLC*22] DELANOY J., LAGUNAS M., CONDOR J., GUTIERREZ D., MASIA B.: A generative framework for image-based editing of material appearance using perceptual attributes. *Computer Graphics Forum 41*, 1 (2022), 453–464. 2, 3, 4, 6, 7, 8, 9, 13

[DLG*20] DELANOY J., LAGUNAS M., GALVE I., GUTIERREZ D., SERRANO A., FLEMING R., MASIA B.: The role of objective and subjective measures in material similarity learning. In *ACM SIGGRAPH 2020 Posters* (New York, NY, USA, 2020), SIGGRAPH '20, Association for Computing Machinery. 2

[DSMG21] DELANOY J., SERRANO A., MASIA B., GUTIERREZ D.: Perception of material appearance: A comparison between painted and rendered images. *Journal of Vision 21*, 5 (05 2021), 16–16. 2

[DTPG11] DONG Y., TONG X., PELLACINI F., GUO B.: Appgen: Interactive material modeling from a single image. In *Proceedings of the 2011 SIGGRAPH Asia Conference* (New York, NY, USA, 2011), SA '11, Association for Computing Machinery. 3

[FDA01] FLEMING R. W., DROR R. O., ADELSON E. H.: How do humans determine reflectance properties under unknown illumination? 2

[FDA03] FLEMING R. W., DROR R. O., ADELSON E. H.: Real-world illumination and the perception of surface reflectance properties. *Journal of Vision 3*, 5 (07 2003), 3–3. 2

[Fle14] FLEMING R. W.: Visual perception of materials and their properties. *Vision Research 94* (2014), 62–75. 2

[FPG01] FERWERDA J. A., PELLACINI F., GREENBERG D. P.: Psychophysically based model of surface gloss perception. In *Human Vision and Electronic Imaging VI* (2001), Rogowitz B. E., Pappas T. N., (Eds.), vol. 4299, International Society for Optics and Photonics, SPIE, pp. 291 – 301. 2

[FS19] FLEMING R. W., STORRS K. R.: Learning to see stuff. *Current Opinion in Behavioral Sciences 30* (2019), 100–108. Visual perception. 2

[FV14] FILIP J., VÁVRA R.: Template-based sampling of anisotropic BRDFs. *Computer Graphics Forum 33*, 7 (October 2014), 91–99. 7

[FWG13] FLEMING R. W., WIEBEL C., GEGENFURTNER K.: Perceptual qualities and material classes. *Journal of Vision 13*, 8 (07 2013), 9–9. 2

[GAA*17] GULRAJANI I., AHMED F., ARJOVSKY M., DUMOULIN V., COURVILLE A. C.: Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems* (2017), Guyon I., Luxburg U. V., Bengio S., Wallach H., Fergus R., Vishwanathan S., Garnett R., (Eds.), vol. 30, Curran Associates, Inc. 4

[GMLMG12] GARCES E., MUNOZ A., LOPEZ-MORENO J., GUTIERREZ D.: Intrinsic images by clustering. *Computer Graphics Forum 31*, 4 (2012). 3

[GPAM*14] GOODFELLOW I., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A., BENGIO Y.: Generative adversarial nets. In *Advances in Neural Information Processing Systems* (2014), Ghahramani Z., Welling M., Cortes C., Lawrence N., Weinberger K., (Eds.), vol. 27, Curran Associates, Inc. 3, 4

[GSLM*08] GUTIERREZ D., SERON F. J., LOPEZ-MORENO J., SANCHEZ M. P., FANDOS J., REINHARD E.: Depicting procedural caustics in single images. *ACM Trans. on Graphics 27*, 5 (dec 2008). 3

[GWA*15] GKIOULEKAS I., WALTER B., ADELSON E. H., BALA K., ZICKLER T.: On the appearance of translucent edges. In *Proc. Computer Vision and Pattern Recognition (CVPR)* (June 2015). 2

[GXZ*13] GKIOULEKAS I., XIAO B., ZHAO S., ADELSON E. H., ZICKLER T., BALA K.: Understanding the role of phase function in translucent appearance. *ACM Trans. on Graphics 32*, 5 (oct 2013). 2

[Hdr] HDRIHAVEN: Hdrihaven. URL: https://www.hdrihaven.com/. 6

[HFM16] HAVRAN V., FILIP J., MYSZKOWSKI K.: Perceptually motivated brdf comparison using single image. *Computer Graphics Forum 35*, 4 (2016), 1–12. 2

[HGC*20] HU B., GUO J., CHEN Y., LI M., GUO Y.: Deepbrdf: A deep representation for manipulating measured brdf. *Computer Graphics Forum 39*, 2 (2020), 157–166. 2

[HHD*22] HU Y., HE C., DESCHAINTRE V., DORSEY J., RUSHMEIER H.: An inverse procedural modeling pipeline for svbrdf maps. *ACM Trans. on Graphics 41*, 2 (jan 2022). 2

[HLM06] HO Y.-X., LANDY M. S., MALONEY L. T.: How direction of illumination affects visually perceived surface roughness. *Journal of Vision 6*, 5 (04 2006), 8–8. 2

[HMP*17] HIGGINS I., MATTHEY L., PAL A., BURGESS C., GLOROT X., BOTVINICK M., MOHAMED S., LERCHNER A.: beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR (Poster)* (2017). 3

[HZK*19] HE Z., ZUO W., KAN M., SHAN S., CHEN X.: Attgan: Facial attribute editing by only changing what you want. *IEEE Transactions on Image Processing 28*, 11 (Nov 2019), 5464–5478. 2, 3, 4

[Jak10] JAKOB W.: Mitsuba renderer, 2010. URL: http://www.mitsuba-renderer.org. 4

[JSR*22] JAKOB W., SPEIERER S., ROUSSEL N., NIMIER-DAVID M., VICINI D., ZELTNER T., NICOLET B., CRESPO M., LEROY V., ZHANG Z.: Mitsuba 3 renderer, 2022. URL: https://mitsuba-renderer.org. 7

[Kal] KALEIDO: Removebg. URL: https://www.remove.bg. 6

[KALL17] KARRAS T., AILA T., LAINE S., LEHTINEN J.: Progressive growing of gans for improved quality, stability, and variation, 2017. 8

[KB14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 6

[KFB10] KŘIVÁNEK J., FERWERDA J. A., BALA K.: Effects of global illumination approximations on material appearance. *ACM Trans. on Graphics 29*, 4 (jul 2010). 2

[KRFB06] KHAN E. A., REINHARD E., FLEMING R. W., BÜLTHOFF H. H.: Image-based material editing. *ACM Trans. on Graphics 25*, 3 (jul 2006), 654–663. 3

[KW13] KINGMA D. P., WELLING M.: Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013). 3

[LAD*21] LACAMBRA J. C., ARTO M. L., DELANOY J., CORCOY B. M., GUTIÉRREZ D., ET AL.: Normal map estimation in the wild. *Jornada de Jóvenes Investigadores del I3A 9* (2021). 7

[LBAD*06] LAWRENCE J., BEN-ARTZI A., DECORO C., MATUSIK W., PFISTER H., RAMAMOORTHI R., RUSINKIEWICZ S.: Inverse shade trees for non-parametric material representation and editing. *ACM Trans. on Graphics 25*, 3 (jul 2006), 735–745. 2

[LCY*17] LIU G., CEYLAN D., YUMER E., YANG J., LIEN J.-M.: Material editing using a physically based rendering network. In *Proc. International Conference on Computer Vision (ICCV)* (2017), pp. 2280–2288. 3

[LDX*19] LIU M., DING Y., XIA M., LIU X., DING E., ZUO W., WEN S.: Stgan: A unified selective transfer network for arbitrary image attribute editing. In *Proc. Computer Vision and Pattern Recognition (CVPR)* (June 2019). 2, 3, 6

[LMJH*10] LOPEZ-MORENO J., JIMENEZ J., HADAP S., REINHARD E., ANJYO K., GUTIERREZ D.: Stylized depiction of images based on depth perception. In *Proc. International Symposium on Non-Photorealistic Animation and Rendering (NPAR)* (New York, NY, USA, 2010), NPAR '10, Association for Computing Machinery, p. 109–118. 3

[LMS*19] LAGUNAS M., MALPICA S., SERRANO A., GARCES E., GUTIERREZ D., MASIA B.: A similarity measure for material appearance. *ACM Trans. Graph. 38*, 4 (jul 2019). 2, 4

[LSGM21] LAGUNAS M., SERRANO A., GUTIERREZ D., MASIA B.: The joint role of geometry and illumination on material recognition. *Journal of Vision 21*, 2 (02 2021), 2–2. 2, 8

[LZU*17] LAMPLE G., ZEGHIDOUR N., USUNIER N., BORDES A., DENOYER L., RANZATO M. A.: Fader networks:manipulating images by sliding attributes. In *Advances in Neural Information Processing Systems* (2017), Guyon I., Luxburg U. V., Bengio S., Wallach H., Fergus R., Vishwanathan S., Garnett R., (Eds.), vol. 30, Curran Associates, Inc. 2, 3, 13

[MB10] MALONEY L. T., BRAINARD D. H.: Color and material perception: Achievements and challenges. *Journal of Vision 10*, 9 (12 2010), 19–19. 2

[MGZ*17] MYLO M., GIESEL M., ZAIDI Q., HULLIN M., KLEIN R.: Appearance bending: A perceptual editing paradigm for data-driven material models. In *Proceedings of the Conference on Vision, Modeling and Visualization* (Goslar, DEU, 2017), VMV '17, Eurographics Association, p. 9–16. 2

[MLMG19] MAO R., LAGUNAS M., MASIA B., GUTIERREZ D.: The effect of motion on the perception of material appearance. In *ACM Symposium on Applied Perception* (New York, NY, USA, 2019), SAP '19, Association for Computing Machinery. 2

[MLTFR19] MAXIMOV M., LEAL-TAIXE L., FRITZ M., RITSCHEL T.: Deep appearance maps. In *Proc. International Conference on Computer Vision (ICCV)* (October 2019). 3

[MPBM03] MATUSIK W., PFISTER H., BRAND M., MCMILLAN L.: A data-driven reflectance model. *ACM Trans. on Graphics 22*, 3 (jul 2003), 759–769. 4, 7

[MST*20] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHI R., NG R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In *Computer Vision – ECCV 2020* (Cham, 2020), Vedaldi A., Bischof H., Brox T., Frahm J.-M., (Eds.), Springer International Publishing, pp. 405–421. 2

[NDM06] NGAN A., DURAND F., MATUSIK W.: Image-driven Navigation of Analytical BRDF Models. In *Symposium on Rendering* (2006), Akenine-Moeller T., Heidrich W., (Eds.), The Eurographics Association. 4

[NJR15] NIELSEN J. B., JENSEN H. W., RAMAMOORTHI R.: On optimal, minimal brdf sampling for reflectance acquisition. *ACM Trans. on Graphics 34*, 6 (oct 2015). 2

[NOGRR21] NIEVES J. L., OJEDA J., GÓMEZ-ROBLEDO L., ROMERO J.: Psychophysical determination of the relevant colours that describe the colour palette of paintings. *Journal of Imaging 7*, 4 (Apr 2021), 72. 2

[PFG00] PELLACINI F., FERWERDA J. A., GREENBERG D. P.: Toward a psychophysically-based light reflection model for image synthesis. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (USA, 2000), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., p. 55–64. 2

[PGM*19] PASZKE A., GROSS S., MASSA F., LERER A., BRADBURY J., CHANAN G., KILLEEN T., LIN Z., GIMELSHEIN N., ANTIGA L., DESMAISON A., KOPF A., YANG E., DEVITO Z., RAISON M., TEJANI A., CHILAMKURTHY S., STEINER B., FANG L., BAI J., CHINTALA S.: Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems* (2019), Wallach H., Larochelle H., Beygelzimer A., d'Alché-Buc F., Fox E., Garnett R., (Eds.), vol. 32, Curran Associates, Inc. 6

[PL07] PELLACINI F., LAWRENCE J.: Appwand: Editing measured materials using appearance-driven optimization. In *ACM SIGGRAPH 2007 Papers* (New York, NY, USA, 2007), SIGGRAPH '07, Association for Computing Machinery, p. 54–es. 3

[RFB15] RONNEBERGER O., FISCHER P., BROX T.: U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (2015), Springer, pp. 234–241. 3

[RRF*16] REMATAS K., RITSCHEL T., FRITZ M., GAVVES E., TUYTELAARS T.: Deep reflectance maps. In *Proc. Computer Vision and Pattern Recognition (CVPR)* (2016). 3

[SAF21] STORRS K. R., ANDERSON B. L., FLEMING R. W.: Unsupervised learning predicts human perception and misperception of gloss. *Nature Human Behaviour 5*, 10 (Oct 2021), 1402–1417. 2

[SCW*21] SERRANO A., CHEN B., WANG C., PIOVARCI M., SEIDEL H.-P., DIDYK P., MYSZKOWSKI K.: The effect of shape and illumination on material perception: model and applications. *ACM Trans. on Graph.* (2021). 7

[SDZ*21] 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 *Proc. Computer Vision and Pattern Recognition (CVPR)* (June 2021), pp. 7495–7504. 2

[SFV20] SCHMIDT F., FLEMING R. W., VALSECCHI M.: Softness and weight from shape: Material properties inferred from local shape features. *Journal of Vision 20*, 6 (06 2020), 2–2. 2

[SGM*16] SERRANO A., GUTIERREZ D., MYSZKOWSKI K., SEIDEL H.-P., MASIA B.: An intuitive control space for material appearance. *ACM Trans. on Graphics 35*, 6 (nov 2016). 2

[TFCRS11] THOMPSON W., FLEMING R., CREEM-REGEHR S., STEFANUCCI J. K.: *Visual perception from a computer graphics perspective*. CRC press, 2011. 2

[VLD07] VANGORP P., LAURIJSSEN J., DUTRÉ P.: The influence of shape on the perception of material reflectance. In *ACM SIGGRAPH 2007 Papers* (New York, NY, USA, 2007), SIGGRAPH '07, Association for Computing Machinery, p. 77–es. 2

[WAKB09] WILLS J., AGARWAL S., KRIEGMAN D., BELONGIE S.: Toward a perceptual space for gloss. *ACM Trans. on Graphics 28*, 4 (sep 2009). 2, 4

[XWT*08] XUEY S., WANG J., TONG X., DAI Q., GUO B.: Image-based Material Weathering. *Computer Graphics Forum* (2008). 3

[XZG*20] XIAO B., ZHAO S., GKIOULEKAS I., BI W., BALA K.: Effect of geometric sharpness on translucent material perception. *Journal of Vision 20*, 7 (07 2020), 10–10. 2

[YS19] YU Y., SMITH W. A. P.: Inverserendernet: Learning single image inverse rendering. In *Proc. Computer Vision and Pattern Recognition (CVPR)* (June 2019). 2

[YX16] YANG J., XIAO S.: An inverse rendering approach for heterogeneous translucent materials. In *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry - Volume 1* (New York, NY, USA, 2016), VRCAI '16, Association for Computing Machinery, p. 79–88. 2

[ZFWW20] ZSOLNAI-FEHÉR K., WONKA P., WIMMER M.: Photorealistic material editing through direct image manipulation. *Computer Graphics Forum 39*, 4 (2020), 107–120. 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. *ACM Trans. on Graphics 40*, 6 (dec 2021). 2

[ZZW*21] ZHENG C., ZHENG R., WANG R., ZHAO S., BAO H.: A compact representation of measured brdfs using neural processes. *ACM Trans. on Graphics 41*, 2 (nov 2021). 2

Appendix A: STU Details

The STU cells are a variant of the GRU [CvMG*14, CGCB14] model and allow encoder-decoder Convolutional Neural Networks (CNNs) to retain the relevant information over a long period. Let's say we want to send the feature map of the l^{th} encoder layer denoted by $\mathbf{f}^l_{enc}$ to the l^{th} layer of the decoder. Given a single STU, as is shown in Figure 2, the hidden state $\hat{\mathbf{s}}^{\mathbf{l+1}}$ holds the information from the previous STU cell of the layer $l+1$. This input is used to remove information from $\mathbf{f}^l_{enc}$ an generate the output feature map $\mathbf{f}^l_t$ as is shown in Figure 2. The hidden state $\mathbf{s}^{\mathbf{l}}$ of the cell is calculated and sent to the next layer $l-1$. The hidden state $\hat{\mathbf{s}}^{l+1}$ also has information of the target attribute value:

$$\hat{\mathbf{s}}^{l+1} = \mathbf{W}_t *_U \left[\mathbf{s}^{l+1}, \mathbf{att}_t\right]. \tag{10}$$

Where $*_U$ and $[\cdot,\cdot]$ denote the upsampling operation followed by a convolution operation, and the concatenation operator respectively. The update gate $\mathbf{u}$ helps the cell to determine how much of the past information (from the previous cell) needs to be passed along to the future, while reset gate $\mathbf{r}$ is used to decide how much of the past information to forget. These tensors are computed as:

$$\mathbf{u}^l = \sigma\left(\mathbf{W}_u * \left[\mathbf{f}_{enc}, \hat{\mathbf{s}}^{l+1}\right]\right), \tag{11}$$

$$\mathbf{r}^l = \sigma\left(\mathbf{W}_r * \left[\mathbf{f}_{enc}, \hat{\mathbf{s}}^{l+1}\right]\right). \tag{12}$$

The sigmoid activation function $\sigma(\cdot)$ is applied to normalize the result between 0 and 1. $\mathbf{W}_u$ and $\mathbf{W}_r$ are the weights matrices updated during training. The hidden state $\mathbf{s}^l$ uses $\mathbf{r}^l$ to store the relevant information from the past and is calculated as follows:

$$\mathbf{s}^l = \mathbf{r}^l \circ \hat{\mathbf{s}}^{l+1}. \tag{13}$$

Where $\circ$ expresses the Hadamard product. This operation between $\mathbf{r}^l$ and $\hat{\mathbf{s}}^{l+1}$, determines what to remove from the previous cell. To compute $\mathbf{f}^l_t$; first, nonlinearity is introduced in the form of *tanh* to ensure that the values in the candidate feature map $\hat{\mathbf{f}}^l_t$ remain in the interval $[-1,1]$:

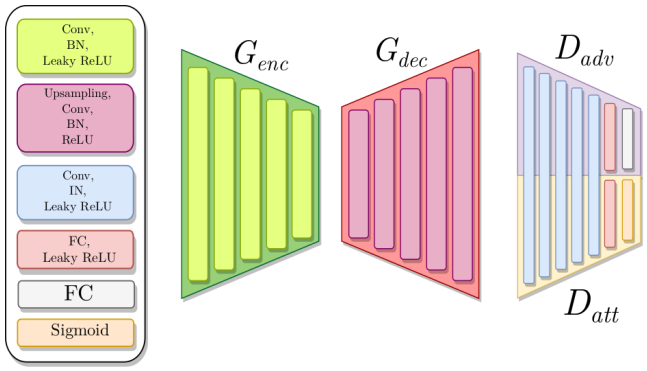

Figure 15: *Architecture of our generator G and discriminator D. BN, IN and FC denote Batch Normalization, Instance Normalization and Fully Connected layer respectively.*

$$\hat{\mathbf{f}}^l_t = tanh\left(\mathbf{W}_h * \left[\mathbf{f}_{enc}, \mathbf{s}^l\right]\right). \tag{14}$$

Finally, the update gate $\mathbf{u}$ is needed to determine what to collect from the candidate feature map $\hat{\mathbf{f}}^l_t$ and $\hat{\mathbf{s}}^{l+1}$, so we compute the Hadamard product as with Equations 11 and 12, last, the result is convolved by the weights $\mathbf{W}_h$:

$$\mathbf{f}^l_t = \left(1-\mathbf{u}^l\right) \circ \hat{\mathbf{s}}^{l+1} + \mathbf{u}^l \circ \hat{\mathbf{f}}^l_t. \tag{15}$$

Appendix B: Additional Details of Our Architecture

Our framework is composed of an encoder-decoder network G and the auxiliary attribute predictor and image discriminator D only used during training. The generator G is composed of an encoder G_{enc} made of 5 convolutional layers that reduce the spatial dimensions of the input image by a factor of two, and a decoder G_{dec} composed of 5 convolutional layers, scaling the input feature of each layer by an upsampling operation. The architecture of the discriminator D is similar to G_{enc} (5 convolutional layers), but both D_{adv} and D_{att} apply full connected layers to output their predictions, as is shown in Figure 15.

Trade-off Parameters Generative models are highly unstable during training, selecting the correct trade-off parameters is crucial. We have observed that the decoder's attribute manipulation loss $\mathcal{L}_{D_{att}}$ decreases rapidly compared to its adversarial loss function $\mathcal{L}_{D_{adv}}$. As we see in Figure 16 increasing λ_2 improves the editing ability of our framework. On the other hand, the decoder applies several transpose convolutions operations to reconstruct the target image. Unfortunately, this architecture may introduce some artifacts in the edited image. We avoid using transpose convolutions in our framework since they produce artifacts on the final image as is shown in Figure 16.

Comparison with the State of the Art The number of trainable parameters is an important factor when a deep learning model is being designed because a large number of parameters involves using

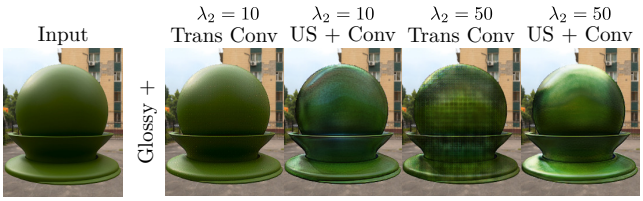

Figure 16: *Ablation studies where we train four versions of our framework. Arrow (pointing up) indicates we set the target attribute glossy* $\mathbf{att}_t$ *to 1. From left to right: edited synthetic image by our framework using transposed convolutions and training with* $\lambda_2 = 10$*, upsampling followed by a convolutional layer training with* $\lambda_2 = 10$*, transposed convolutions, and training with* $\lambda_2 = 50$ *and upsampling followed by a convolutional layer training with* $\lambda_2 = 50$*.*

a lot of memory to train the models and, often, resources are scarce. The work from Delanoy et al. [DLC*22] propose a framework composed of two generators G_i with $i \in \{1,2\}$ based on a Fader Network [LZU*17] architecture. G_1 compress the low-resolution input image of the single object in a latent code $\mathbf{z}_1$. G_2 replicates this behavior but takes high-resolution images as input to generate its latent code $\mathbf{z}_2$. Since both latent codes, $\mathbf{z}_i$ must not contain perceptual information of the high-level perceptual attribute a, during training both generators G_i play an adversarial game with a latent discriminator LD_i. Also, the authors introduce an image discriminator D that plays an adversarial game with G_2 to enhance the quality of edited images. Table 2 shows the trainable parameters of the whole framework.

Our framework is composed of one generator G and one discriminator D, while the previous approach needs two generators $\mathcal{G}_i$ helped by other two latent discriminators $\mathcal{LD}_i$ and one image discriminator $\mathcal{C}/\mathcal{D}$ to improve the editing ability. In Table 3 we give the number of total trainable parameters of our framework. Since our framework is simpler than the previous one, it has fewer trainable parameters, so we reduce notably memory usage by reducing the trainable parameters from 58 920 489 parameters to 33 326 314 and thus saving 43% of memory.

Table 2: *Number of trainable parameters per module for the previous method.*

Module	Trainable Parameters
G_1	20 356 515
LD_1	4 326 401
G_2	4 610 179
LD_2	14 813 697
C/D	14 813 697
Total Parameters	58 920 489

Table 3: *Number of trainable parameters per module for our framework.*

Module	Trainable Parameters
G ($G_{enc} + G_{dec} + G_{st}$)	13 758 280
D (both D_{att} and D_{adv})	19 568 034
Total Parameters	33 326 314