Sun Off, Lights On: Photorealistic Monocular Nighttime Simulation for Robust Semantic Perception

Konstantinos Tzevelekakis[1], Shutong Zhang[2], Luc Van Gool[1,3,4], and Christos Sakaridis[1]

[1]ETH Zürich, [2]University of Toronto, [3]KU Leuven, [4]INSAIT

Abstract

Nighttime scenes are hard to semantically perceive with learned models and annotate for humans. Thus, realistic synthetic nighttime data become all the more important for learning robust semantic perception at night, thanks to their accurate and cheap semantic annotations. However, existing data-driven or hand-crafted techniques for generating nighttime images from daytime counterparts suffer from poor realism. The reason is the complex interaction of highly spatially varying nighttime illumination, which differs drastically from its daytime counterpart, with objects of spatially varying materials in the scene, happening in 3D and being very hard to capture with such 2D approaches. The above 3D interaction and illumination shift have proven equally hard to model *in the literature, as opposed to other conditions such as fog or rain. Our method, named Sun Off, Lights On (SOLO), is the first to perform nighttime simulation on single images in a photorealistic fashion by operating in 3D. It first explicitly estimates the 3D geometry, the materials and the locations of light sources of the scene from the input daytime image and relights the scene by probabilistically instantiating light sources in a way that accounts for their semantics and then running standard ray tracing. Not only is the visual quality and photorealism of our nighttime images superior to competing approaches including diffusion models, but the former images are also proven more beneficial for semantic nighttime segmentation in day-to-night adaptation. Code and data are publicly available at* https://github.com/ktzevel/SOLO.

1. Introduction

A key requirement for level-5 automated driving systems and other outdoor autonomous agents is the robust visual perception of the surrounding scene, so that the content of the scene can be parsed under any visual condition [30]. However, the ubiquitous condition of night time has a detrimental effect on the quality of camera measurements due to effects such as underexposure, overexposure, and motion blur [50]. This low input quality at night translates to a drastic deterioration in the performance of semantic perception algorithms for central tasks such as semantic segmentation, as compared to normal conditions or even other adverse conditions such as fog, rain, or snow [35]. What makes things worse is the increased difficulty in the manual pixel-level semantic annotation of real nighttime images due to the above effects, which leads to errors and reduced image coverage in ground-truth annotations, in turn with negative impact on the reliability of models trained on such data.

As a result, a widely used paradigm for robust semantic nighttime segmentation is unsupervised domain adaptation (UDA) from day to night [7, 9, 16, 31, 34–36, 45, 46]. In this paradigm, both labeled – thanks to easier acquisition and annotation – daytime, or source-domain, images and unlabeled nighttime, or target-domain, images are available at training. A core element of such UDA methods is input-level adaptation [14, 22, 39] of source-domain images to the style of the target domain, so that the labels which are inherited by these adapted source-domain images can constrain the semantic segmentation model more effectively on closely-resembling real target-domain images. The three main approaches to such input-level adaptation are physically-based domain translation, learned image translation, and hand-crafted domain transformation.

On the one hand, learned, data-driven models for translating an input image, based e.g. on generative adversarial networks [54] or diffusion models [51], can implicitly capture the statistics and patterns in the source and target domain and have proven very successful for input-level adaptation in synthetic-to-real UDA [14]. However, images adapted with such approaches are not photorealistic, as the latter do not model illumination, which changes drastically from day to night, or the properties of the scene, i.e. its 3D geometry and materials, which affect the spatially-varying interaction of light with the scene at night time and the resulting appearance of the image. The same limitation applies for hand-crafted methods for domain transformation, such as Fourier-based adaptation [48]. On the other hand, while physically-based approaches for domain translation do not require training or reference-style images and have

enjoyed remarkable success in condition-level UDA in the cases of fog [2, 11, 33], rain [12], and snowfall [10], no such approach has been proposed for the ubiquitous nighttime condition to the best of our knowledge.

In this paper, we present the first physically-based monocular approach to nighttime simulation on real daytime images, aiming to afford photorealistic synthetic nighttime counterparts. We pursue this through inverse rendering, probabilistic light source instantiation, and physically-based rendering via ray tracing. Our rationale for photorealistic nighttime simulation is to (i) estimate the scene representations required for ray tracing from the input daytime image, notably the positions of all inactive light sources in the scene, (ii) modify the lighting of the scene by removing the sun and probabilistically activating the aforementioned light sources in a semantics-aware fashion (hence the name of our method Sun Off, Lights On or SOLO), and (iii) relight the scene by running ray tracing with the updated, nighttime lighting to render the nighttime image.

The key novel contributions of SOLO are (i) the semantics-aware probabilistic light source instantiation for shifting the lighting of the scene from day time to night time, (ii) a carefully crafted, normals- and semantics-guided optimization for depth map refinement within the mesh-based 3D reconstruction of our monocular inverse rendering module, as well as (iii) our overall physically-based monocular nighttime simulation pipeline which elegantly combines inverse rendering and ray-tracing-based relighting. Our synthetic nighttime images match the appearance of real nighttime images better thanks to their photorealism and thus serve as a good proxy for the real nighttime domain. We verify this superiority of SOLO in an objective fashion, by using it as the input-level adaptation module of a state-of-the-art UDA pipeline [15] for day-to-night semantic segmentation adaptation on the challenging ACDC-Reference→ACDC-night [37] benchmark, and by employing the Kernel Inception Distance (KID), known to correlate with human judgment [3].

2. Related Work

Scene relighting is a fundamental task in computer vision and graphics. In the context of *novel view synthesis*, it refers to rendering a scene from different camera views [8, 24, 41]. Another version of scene relighting involves rendering a scene for another time of day or under varying but known lighting conditions [41, 43, 52]. Our setting is a special case of the latter, performing photorealistic nighttime simulation by considering ambient lighting, light sources activated at night, and their interactions with the scene.

Relighting through inverse rendering. Conventionally, relighting a scene requires accurate estimation of geometry and material parameters, a process known as inverse rendering. The most prevalent approach is to learn priors on the shape, illumination and reflectance, using large labeled image datasets for geometry and materials like in [23, 38, 40, 44, 49, 52]. Scene relighting is then achieved by forward rendering. SOLO is closely related to these methods by treating state-of-the-art inverse rendering models as black boxes to estimate both material and geometry parameters. Unlike the aforementioned works, SOLO also employs a semantic light source segmentation model, enabling semantically aware explicit reasoning on both the activation and color properties of the light sources. Since the emergence of *neural radiance fields* (NeRFs), a new approach to scene relighting has gained traction. Although the seminal work on NeRFs [25] did not handle relighting, recent works [8, 24, 41, 43] have reformulated the continuous volumetric function to accommodate it. An additional feature of NeRF-based approaches is the ability to recover full 3D models using a sparse set of multi-view images as input. However, in our setting, only one image per scene is available, making NeRF-based approach not easily applicable.

Day-to-night transfer in 2D. A large body of literature, simulates nighttime by employing generative models for *style transfer*. These models are either based on the generative adversarial network (GAN) architecture [5, 17, 54], or on the more recent diffusion architecture [51]. However, during the day-to-night translation, given only a 2D daytime image, these purely data-driven models struggle to account for the activation of light sources, the 3D interactions of light rays with objects in the scene, the rendering of spatially varying illumination, and the simulation of under- or over-exposure. Therefore, even recent diffusion-based architectures [51] cannot accurately simulate nighttime conditions, as evidenced in our experiments. Notable *hand-crafted* 2D-based approaches also exist. In [28], a framework processes a given daytime image by introducing artificial light sources sampled from a nighttime illuminants dataset. Additionally, in [48], a method for UDA is presented, based on the Fourier Transform, which reduces the shift in appearance from the source to the target image by swapping the low-frequency components of the source magnitude spectrum with those from the target magnitude spectrum. This allows the source images to adopt the global appearance characteristics (e.g., texture, lighting conditions). However, both methods strictly operate in the 2D space and fail to provide photorealistic results.

3. Sun Off, Lights On

The proposed nighttime simulation method, named "Sun Off, Lights On" (SOLO), is based on a *single* daytime input image as shown in Fig. 1. SOLO estimates the geometry and materials of the scene through inverse rendering (Sec. 3.1). The novel probabilistic light source instantiation module of SOLO (Sec. 3.2) determines the lighting con-

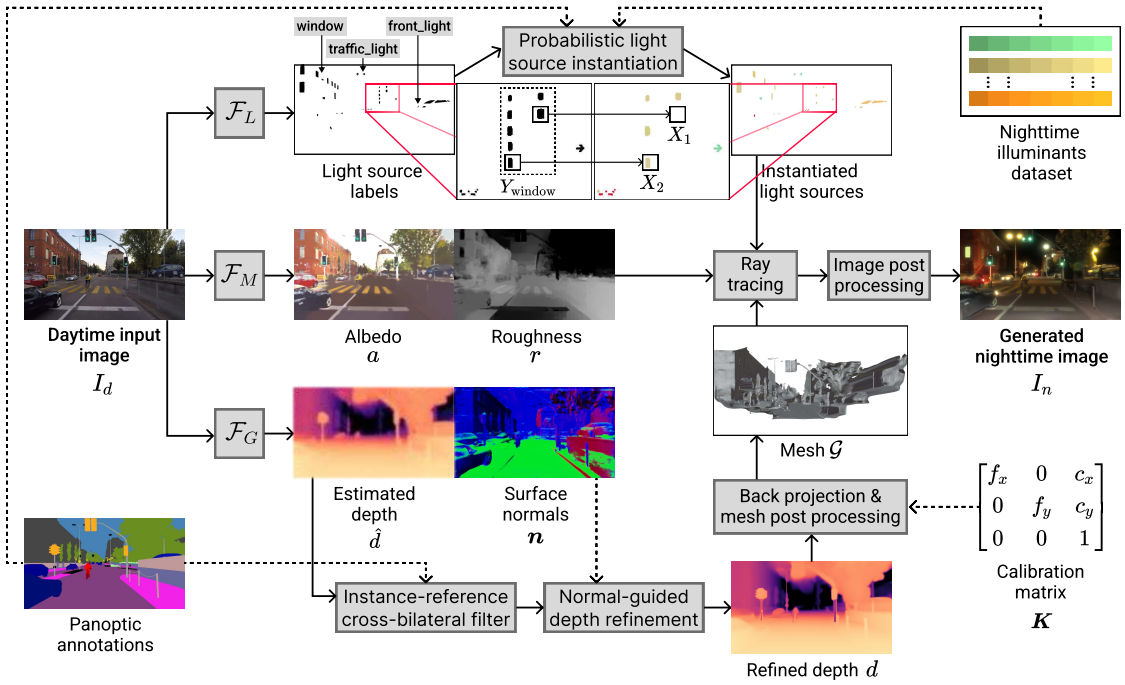


Figure 1. **Overview of SOLO.** Our method accepts as input a single daytime image I_d. Geometric $(\hat{d}, \boldsymbol{n})$ and material (a, r) representations are estimated with the inverse rendering networks $\mathcal{F}_G$ and $\mathcal{F}_M$, respectively. A light source segmentation network $\mathcal{F}_L$ predicts the regions in I_d which correspond to inactive light sources that may emit light at night. The initial depth map $\hat{d}$ is filtered and optimized with guidance from an instance semantic segmentation mask and the estimated surface normal map $\boldsymbol{n}$, respectively. The refined depth map d and the camera intrinsics are used to construct the 3D scene mesh $\mathcal{G}$. Nighttime light sources in the scene are instantiated probabilistically group-wise, using the predictions of $\mathcal{F}_L$ to sample their activation variables $X_i \sim \text{Bernoulli}(Y_c)$, where $Y_c \sim \text{Uniform}(\alpha_c, \beta_c)$ for class c, and the nighttime illuminants dataset to set their chromaticities. The activated light sources, the materials (a, r) and the 3D mesh $\mathcal{G}$ are finally fed to the ray tracing module which renders a raw image that is subsequently post-processed to compute the output nighttime image I_n.

figuration of the nighttime scene. Light sources of different categories are first semantically segmented and grouped and then probabilistically activated to simulate nighttime lighting. Finally, the forward rendering module (Sec. 3.3) uses the estimated geometry, materials, and lighting to perform physically-based rendering (PBR) and thus deliver a photo-realistic nighttime image of the scene.

3.1. Inverse Rendering

3.1.1 Geometry and Materials Estimation

State-of-the-art off-the-shelf monocular estimation networks are employed for both geometry and materials. Specifically, we consider depth $\hat{d}$ and surface normals $\hat{\boldsymbol{n}}$ maps as the dense geometric representations of the scene which are estimated by the geometry network as $F_G(I_d) = (\hat{d}, \hat{\boldsymbol{n}})$. As per the material representations, these are diffuse albedo a and specular roughness r, they are estimated by the material model as $F_M(I_d) = (a, r)$.

As SOLO is applied to daytime *outdoor* scenes, it requires inverse rendering networks trained on such scenes. To the best of our knowledge, no real-world outdoor dataset with dense annotations for materials, in particular for roughness r, exists. However, material properties are not a priori correlated with their occurrence in an indoor or outdoor scene. Thus, the implicit assumption made in the general form of our method is that materials output by an indoor-trained network are accurate for outdoor scenes as well. On the other hand, there is a plethora of geometric models trained on real-world outdoor sets. An important requirement stemming from ray tracing and PBR is the metric character of the reconstructed scene, so the depth units must be known. Thus, only network architectures which output metric depth maps are relevant for SOLO. Since the aforementioned models typically output maps of lower resolution than the original daytime input image, upsampling is required both for geometry and material parameters. Standard bilinear interpolation is used for the geometric maps and joint bilateral upsampling [20] for the material maps. The latter utilizes the corresponding daytime image as reference, performing better than plain bilinear interpolation.

3.1.2 Depth Refinement

Instance-Reference Cross-bilateral Filter. Even a slight misalignment between an actual object boundary and the corresponding depth edge in the prediction $\hat{d}$ of the geometry network F_G deteriorates the realism of the subsequent

3D reconstruction. To remedy this, we adapt the dual-reference cross-bilateral filter of [32]. Apart from spatial information, this filter originally exploits both a color and a semantic reference signal to refine a transmittance map akin to depth. For our depth filtering case, local variations in the color in I_d do not necessarily correspond to variations in depth values. We thus drop the color reference of [32] and only use its semantic reference, replacing the semantic reference labels in [32] with instance-level semantic reference labels. That properly preserves depth edges between different objects of the same semantic class which are adjacent to each other. To formulate our instance-reference cross-bilateral filter, we use $\boldsymbol{p}$ to denote any non-boundary pixel in the depth map $\hat{d}$, and $\boldsymbol{q}$ to denote a pixel belonging to the neighborhood $\mathcal{N}$ of $\boldsymbol{p}$. The filtered depth $\tilde{d}$ at a pixel location $\boldsymbol{p}$ is computed as a weighted average of initial depth values $\hat{d}$:

$$\tilde{d}(\boldsymbol{p}) = \frac{\sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} G_{\sigma_s}(\|\boldsymbol{q}-\boldsymbol{p}\|)\delta(h(\boldsymbol{q})-h(\boldsymbol{p}))\hat{d}(\boldsymbol{q})}{\sum_{\boldsymbol{q}\in\mathcal{N}(\boldsymbol{p})} G_{\sigma_s}(\|\boldsymbol{q}-\boldsymbol{p}\|)\delta(h(\boldsymbol{q})-h(\boldsymbol{p}))}, \tag{1}$$

where G_{σ_s} is the spatial Gaussian kernel applied on the l_2 distance between pixels. This is done only when the instance labels h of the corresponding pixels match, as dictated by the Kronecker delta term δ.

Uncertain Depth Regions. These are regions located near object boundaries, where accurate depth prediction is challenging. Boundaries of thin objects such as people and traffic signs, are typical associated examples. To locate those regions, a sliding window approach is employed, with a $k \times k$ window. All pixels contained in a window are marked as *uncertain* only if both of the following criteria are satisfied: (i) at least two semantic segments overlap with the window, and (ii) the variance of depth values across the window is larger than a predefined threshold t. Finally, objects that are located further from the camera than a distance threshold r are disregarded.

Surface-Normal-Guided Depth Optimization. The surface normals $\boldsymbol{n}$ output by the geometry network F_G provide additional fine-grained geometric information for accurate mesh-based 3D reconstruction of the input scene, on top of depth. In this work, we devise a novel optimization method which exploits this information from normals to refine the depth map. We start by modeling how surface normals $\boldsymbol{n}$ can be inferred from a corresponding depth map $z(x, y)$, assuming a standard pinhole camera model where $u = \frac{f_x}{z}x + c_x$ and $v = \frac{f_y}{z}y + c_y$ are pixel-space coordinates, x and y are 3D camera-frame coordinates, z is the depth at (x, y), f_x and f_y are the focal lengths, and c_x and c_y are the principal point coordinates. The surface normal vector $\boldsymbol{n}$ is perpendicular to the tangent plane of the 3D surface at $(x, y, z(x, y))$. To obtain the direction vector $\boldsymbol{s}$ of this plane, we use the graph function $\boldsymbol{\mathcal{F}}(x, y) = (x, y, z(x, y))$ of z, and its gradient $\nabla\boldsymbol{\mathcal{F}}(x, y) = (\frac{\partial\boldsymbol{\mathcal{F}}}{\partial x}, \frac{\partial\boldsymbol{\mathcal{F}}}{\partial y})^{\mathsf{T}}$. The direction vector $\boldsymbol{s}$ of the tangent plane is perpendicular to both rows of $\nabla\boldsymbol{\mathcal{F}}$, so

$$\boldsymbol{s} = \frac{\partial\boldsymbol{\mathcal{F}}}{\partial x} \times \frac{\partial\boldsymbol{\mathcal{F}}}{\partial y} = (-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1). \tag{2}$$

Finally, the surface normal $\boldsymbol{n}$ is obtained by normalizing $\boldsymbol{s}$. Our optimization loss $\mathcal{L}$ is defined as a weighted sum of two terms. The first term, L_1, is formulated as:

$$L_1 = \frac{1}{m}\sum_{\boldsymbol{p}} \|\nabla\boldsymbol{\mathcal{F}}(\boldsymbol{p}) - \hat{\boldsymbol{n}}(\boldsymbol{p})\|_2^2 \bar{U}(\boldsymbol{p}), \tag{3}$$

where $\hat{\boldsymbol{n}}$ are the surface normals initially predicted by F_G, $\bar{U}$ is the complement of the binary *uncertain* depth region mask, and m is the total number of pixels. The role of $\bar{U}$ is to ignore depth discontinuities in L_1. Minimizing L_1 modifies the depth map $\tilde{d}$ to better conform to the independently predicted normals $\hat{\boldsymbol{n}}$, yielding a more faithful 3D mesh. To avoid smoothing out salient depth details completely and to balance the effect that potential inaccuracies in predicted normals $\hat{n}$ have, the second term of our optimized loss, L_2, quantifies the error between the depth map z which is under optimization and the filtered depth map $\tilde{d}$ as:

$$L_2 = \frac{1}{m}\sum_{\boldsymbol{p}} \Big(\tilde{d}(\boldsymbol{p}) - z(\boldsymbol{p})\Big)^2. \tag{4}$$

The complete optimization loss is $\mathcal{L} = \lambda_1 L_1 + \lambda_2 L_2$, where λ_1 and λ_2 are the weights for the corresponding loss terms. Note that z is initialized with the filtered depth map $\tilde{d}$. We denote the final, optimized depth map by d.

3.1.3 Backprojection and Mesh Post-processing

To initialize the 3D mesh $\mathcal{G}$ to be used subsequently for ray tracing, we use the backprojection equation $\boldsymbol{x} = d\boldsymbol{K}^{-1}\bar{\boldsymbol{p}}$, where $\bar{\boldsymbol{p}} = (u, v, 1)^{\mathsf{T}}$ denotes the homogeneous representation of the pixel coordinates, d is the final depth map, $\boldsymbol{x}$ is the 3D scene point serving as a vertex of $\mathcal{G}$, and $\boldsymbol{K}$ is the calibration matrix. Although this mesh is faithful to the geometry of the 3D scene from the camera's point of view, the monocular information it captures results in geometric errors with spurious faces for occluded objects or objects outside the field of view, creating erroneous shadows at ray tracing. To mitigate this, we apply additional post-processing to remove these faces and subsequently restore a watertight mesh.

3.2. Probabilistic Light Source Instantiation

To illuminate a 3D scene photorealistically, semantic information specific to its light sources is required to assign light source attributes via stochastic rules, along with a

dataset of realistic chromaticities and strengths of nighttime illuminants.

Light source segmentation dataset and model. To the best of our knowledge, no outdoor dataset with light source annotations exists. We first define a novel, comprehensive light source taxonomy for outdoor scenes building upon the object taxonomy of Cityscapes [6], with 13 main light source categories. We then annotate a reasonably-sized daytime set with pixel-level light source labels for this taxonomy by segmenting active *and* inactive light sources in its images. We finally fine-tune a normal semantic segmentation model on this labeled set, using a new prediction layer to account for the different taxonomy. The resulting light source segmentation model $\mathcal{F}_L$ predicts light source labels for the complete daytime set on which we apply SOLO.

Nighttime illuminants dataset. The color appearance of a nighttime illuminant can be specified in terms of the xyY color space. Our nighttime illuminants dataset $\mathcal{N}$ consists of real-world chromaticity samples for each light source category, collected using a gray card and a DSLR camera, following [28]. Each sample includes a raw image of the gray card, illuminated by an instance of the sampled light source category. To avoid pollution from neighboring sources, only the light source to be sampled was visible from the surface of the gray card during collection. The captured raw images are processed with a standard camera pipeline to obtain chromaticity [28, 29, 42]. The illuminants strengths are sampled from empirically defined intervals.

Light source instantiation module. Our probabilistic instantiation module assigns attributes to light sources of the scene based on stochastic rules, conditioned by semantic and instance information. This information is incorporated in two ways: (i) by leveraging the light source label, such as "vehicle front light", and (ii) by exploiting the instance-level semantic label, such as "car 2". By combining these attributes, a tree structure is constructed, which specifies the light source group that a light source belongs to. More specifically, each node of this tree can either correspond to (i) a light source group or (ii) a light source/leaf node, whereas the edges of the tree indicate membership. For example, two "vehicle front lights" can both be children nodes of the same "car" light source group. Three attributes are assigned to each light source: the chromaticity, the strength, and the probability of activation y. These attributes primarily depend on the light source category. Moreover, it is plausible for light sources belonging to the same group, e.g. the front left and front right lights of a car, to share the same attributes. Light source attributes are modeled as random variables. In particular, chromaticity follows a discrete uniform distribution over the relevant samples of $\mathcal{N}$, whereas probability of activation and strength follow continuous uniform distributions over empirically defined intervals. In particular, the stochastically sampled probability of activation y is in turn used to define another Bernoulli variable that models the actual activation of the light source. That is, the random activation variable X for a light source follows $X \sim \text{Bernoulli}(y)$, where the Bernoulli parameter y is the realization of the intermediate variable $Y \sim \text{Uniform}(\alpha, \beta)$. More details are available in Appendix D of the supplement.

3.3. Forward Rendering

All constituents of the scene are combined via forward rendering. The input is the mesh $\mathcal{G}$ overlaid with the estimated materials and active light sources, as shown in Fig. 1. Moreover, head lights from the ego-vehicle are simulated to enhance realism. We run ray tracing to generate a linear image which is later fed to a standard post-processing pipeline resulting in a photorealistic nighttime image.

Physically-based rendering and ray tracing is formulated using the reflectance equation:

$$\begin{aligned} L_o(\boldsymbol{x}, \boldsymbol{\omega}_o) &= L_e(\boldsymbol{x}, \boldsymbol{\omega}_o) + L_r(\boldsymbol{x}, \boldsymbol{\omega}_o), \\ L_r(\boldsymbol{x}, \boldsymbol{\omega}_o) &= \int_\Omega f_r(\boldsymbol{x}, \boldsymbol{\omega_o}, \boldsymbol{\omega_i}) L_i(\boldsymbol{x}, \boldsymbol{\omega}_i)(\boldsymbol{\omega}_i \cdot \boldsymbol{n}) d\boldsymbol{\omega}_i, \end{aligned} \tag{5}$$

where for a point $\boldsymbol{x}$ on a surface, L_o, L_e, and L_r denote the outgoing, emitted, and reflected radiance, respectively. Moreover, $\boldsymbol{\omega}_o$ and $\boldsymbol{\omega}_i$ correspond to outgoing and incident light directions respectively, $\boldsymbol{n}$ is the normal vector, and f_r is the bidirectional reflectance distribution function (BRDF). In particular, we employ the physically motivated "Disney" BRDF proposed in [4] and later adopted by Unreal Engine 4 [18]. This BRDF is formulated as

$$f_r(\boldsymbol{x}, \boldsymbol{\omega_i}, \boldsymbol{\omega_o}) = f_d(\boldsymbol{x}, \boldsymbol{\omega_i}, \boldsymbol{\omega_o}) + f_s(\boldsymbol{x}, \boldsymbol{\omega_i}, \boldsymbol{\omega_o}), \tag{6}$$

where f_d and f_s are the diffuse and specular BRDF components. For ray tracing, we also model the directionality of the light sources, so that both strongly directional and rather diffuse light sources can be simulated. To this end, the strength of a light source is weighted by a function g of the incident direction $\boldsymbol{v}$ and of the normal vector $\hat{\boldsymbol{n}}$ of the *area light source* surface, formulated as $g(\boldsymbol{v}, \hat{\boldsymbol{n}}) = \cos(\pi(|\boldsymbol{v} \cdot \hat{\boldsymbol{n}}| - 1)/2)$.

Image post-processing. We employ a post-processing pipeline imitating the steps of a standard image signal processor (ISP). This serves two purposes. On the one hand, the appropriate transformations should be applied to make the generated image displayable. On the other hand, since our nighttime images are meant for inputs of neural networks, the visual artifacts present in real nighttime images should also appear in our simulated images to minimize the distribution shift. Our post-processing pipeline starts by adjusting the brightness of the image, setting the exposure appropriately. Since the image is in the linear XYZ color space, the standard Bradford color adaptation method [21]

along with gamma correction are used to transform the image to the sRGB color space. Moreover, fog glare is incorporated to make the appearance of light sources more realistic and noise is added using the standard heteroscedastic Gaussian noise model, following [28].

4. Experiments

4.1. Implementation Details

The state-of-the-art pre-trained model of [23] is employed as the materials estimation network F_M. However, this model has limitations. First, its indoor training data do not include several materials commonly found in outdoor scenes. Second, these training data only include dielectric materials, leading to low-quality material maps, especially in regions with *metallic* objects. We experimented with more recent indoor trained models, such as that of [53], but they also estimated materials poorly. We attribute these shortcomings to the large distribution shift between the materials in indoor training sets and those in real-world outdoor scenes. As a result, the roughness estimates are not sufficiently accurate. Notably, the specular microfacet term f_s in (6) is very sensitive to the roughness value. Thus, in all our experiments, we revert to using only the diffuse BRDF component f_d in (6). We use the state-of-the-art pre-trained UniDepth [27] and iDisc [26] networks to predict depth and surface normals, respectively. For depth refinement, σ_s is set to 5px in (1). For uncertain depth regions, we set k=10px, t=0.01, and r to the mid-range of the scene's depth. Normal-guided depth optimization uses Adam [19] for 1000 iterations with a learning rate of 2e-4. We set λ_1=50 and λ_2=1 in this optimization. The camera intrinsics are f_x=f_y=1780px, c_x=959.5px and c_y=539.5px. For PBR via ray tracing, we adopt the multi-scattering GGX implementation [13] of the Cycles path tracer [1], providing off-the-shelf physically based results. In the post-processing pipeline, we set exposure to 3.25 stops, gamma to 2.2, and employ the implementation of [28] for noise addition.

4.2. Datasets

In our experiments, we utilize images from ACDC [37], which provides panoptic annotations of the 19 Cityscapes [6] evaluation classes for 4006 images. ACDC includes a nighttime split, further divided into training, validation and test sets. Moreover, ACDC includes daytime, clear-weather counterparts for 1003 images in the training and validation split, referred to as ACDC-Reference. We focused on ACDC for evaluating SOLO since its reference split includes geographically aligned, annotated daytime counterparts of nighttime images, captured with the same camera, making it ideal for day-to-night UDA, as source and target domains only differ by the time of day. Moreover, due to resource limitations, annotating a large and diverse image set with light sources was infeasible, so our trained light source segmentation model may not generalize equally well to images from different sets.

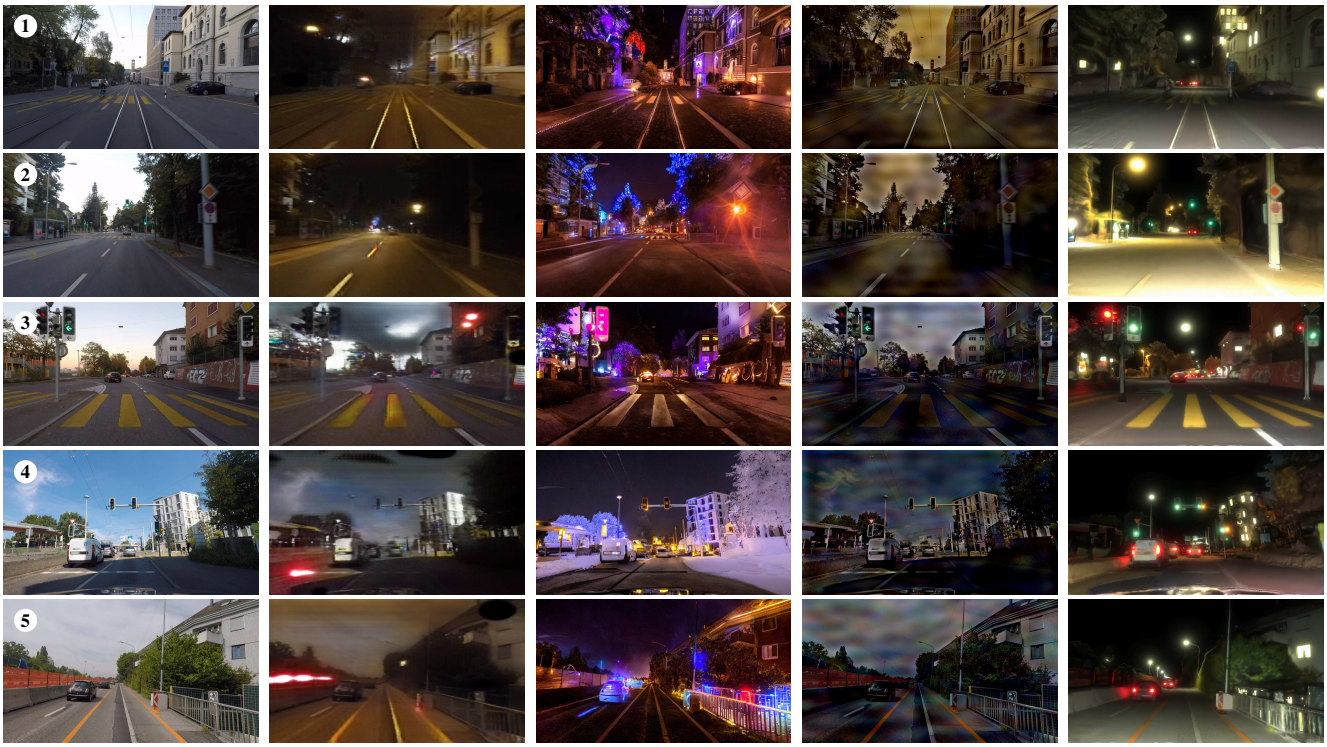
ACDC Light Sources is a set contributed by this work, containing panoptic annotations of active and inactive light sources for ACDC-Reference. An initial set of 350 images was annotated manually and a light source segmentation model was then utilized for the rest 653 of the images. To this end, we fine-tuned a SegFormer model [47] for 40K steps using 320 annotated images as the training set. More details and samples are available in Appendix A of the supplement.

Nighttime Illuminants are derived from 60 images of a gray card illuminated by different outdoor nighttime light sources. There are five samples on average per each of the 12 light source categories of the dataset, and each sample corresponds to a chromaticity value as described in Sec. 3.2. More details and samples are available in Appendix C of the supplement.

Evaluation dataset. Images in ACDC-Reference are used in the evaluation. The UDA pipeline of HRDA [15] for semantic segmentation is employed. With the target and source domains corresponding to night time and day time respectively, a *source dataset* is formed from ACDC-Reference, with 800 training and 203 validation images. Moreover, a *target dataset* is formed from ACDC-night, with 400 training and 106 validation images. The 500 test images of ACDC-night with withheld labels are used as test set. The predictions of all methods on this test set are submitted to the public ACDC benchmark for evaluation.

4.3. Comparisons to The State of The Art

SOLO is compared against other state-of-the-art stylization methods, including Fourier Domain Adaptation (FDA) [48], ControlNet [51] and CycleGAN [54]. For FDA, a bandwidth of 0.01 is set. For ControlNet, the prompt 'transform this image to nighttime' is used, showing little difference when paraphrased. For the *qualitative comparison*, the generated stylized images alongside their daytime inputs are displayed in Fig. 2. In column (b), CycleGAN generations exhibit several issues including: the incomplete removal of the daytime ambient illumination (images 3 and 4), the unrealistic, light blue, appearance of the nighttime sky, the spatially inconsistent red light glows (images 3, 4 and 5), the inactive vehicle and traffic lights, and the unrealistic illumination of regions that surround activated light sources as though those sources were inactive. By contrast, ControlNet (column (c)) achieves a more realistic rendering of the nighttime sky. However, traffic (images 3 and 4) and street (images 2 and 5) lights remain inactive, similar to CycleGAN. Additionally, a strong violet tint is present across the stylized images, and the surface tex-



(a) Input image (b) CycleGAN (c) ControlNet (d) FDA (e) SOLO

Figure 2. **Qualitative comparison of day-to-night translation methods.** From left to right: daytime input images, and synthesized nighttime results of CycleGAN [54], ControlNet [51], FDA [48], and SOLO (ours). More samples available in Appendix B of the supplement.

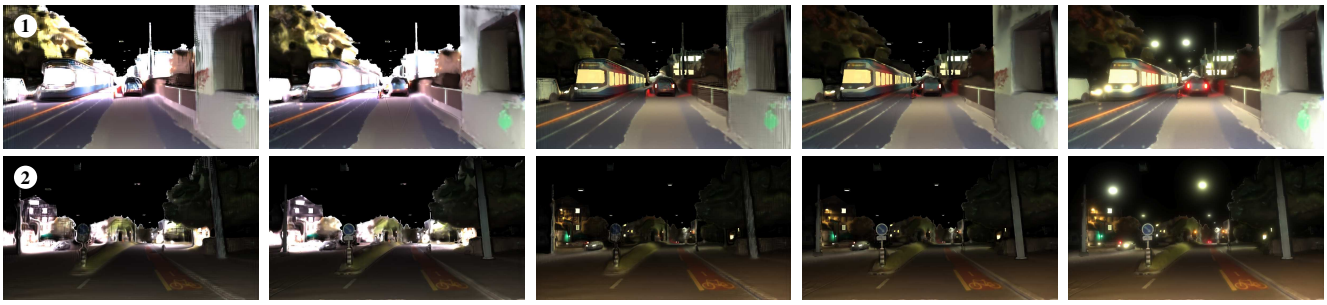
Table 1. **Comparison of day-to-night translation methods (ACDC-Reference→ACDC-night) using the HRDA UDA framework for the semantic segmentation evaluation and the Kernel Inception Distance (KID) for realism evaluation.** The HRDA is evaluated on the test split of ACDC-night. The KID is calculated between the stylized ACDC-Reference images and real ACDC-night images.

Stylization	road	sidew.	build.	wall	fence	pole	light	sign	veget.	terrain	sky	person	rider	car	truck	bus	train	motorc.	bicycle	**mIoU**	**KID***
None	95.8	78.6	83.1	51.6	37.7	56.8	52.2	57.2	72.4	46.4	80.8	66.0	36.2	81.8	18.6	47.8	88.1	51.8	48.4	60.6	n/a
With CycleGAN	96.2	79.8	82.4	46.2	36.7	55.3	55.9	58.0	68.5	47.5	78.3	66.3	36.4	83.3	43.1	53.2	88.9	53.2	52.1	62.2	0.076
With ControlNet	95.2	76.9	80.2	45.0	27.2	53.2	55.7	57.9	54.2	46.0	66.2	64.0	39.4	81.3	56.0	50.3	88.8	49.9	47.1	59.7	0.117
With FDA	96.3	80.6	83.0	45.0	35.3	57.8	56.3	61.0	69.1	48.7	78.6	67.6	33.4	84.8	42.5	67.5	90.3	52.2	54.3	63.4	0.207
With SOLO (Ours)	95.7	78.8	82.7	49.4	31.2	53.7	51.6	56.8	71.2	47.6	78.6	64.3	36.1	83.2	65.1	62.3	89.3	50.9	48.5	63.0	0.056

*lower is better.

tures are not inherited from the daytime inputs. FDA (column (d)) faces similar challenges to CycleGAN, including the incomplete elimination of the ambient light, the unrealistic color of the nighttime sky, a repetitive gray pattern in sky regions, and the failure to account for light sources, resulting in an unrealistic nighttime result. On the other hand, SOLO (column (e)) tackles most of the aforementioned issues. The ambient illumination from the daytime image is eliminated, scene illumination fully depends on the activated light sources, and the surface textures from the daytime image are closely resembled. Notably, the instantiation of the light sources is explicitly handled. The colors of the lights are sampled from the nighttime illuminants dataset, conditioned to the inherited daytime semantics. The noise addition and fog glare effects realistically simulate various typical nighttime artifacts.

SOLO is quantitatively evaluated on semantic segmentation using the HRDA [15] UDA framework. In particular, the 'MiT-B5' SegFormer model [47], pre-trained on daytime images, is adapted to the target nighttime domain. Moreover, three random seeds are used for each setting during training. The mean intersection-over-union (mIoU) is used to select the best model for evaluation on the test set of ACDC-night. The test mIoU and class-level IoUs are reported. In Table 1, SOLO outperforms both state-of-the-art input-level adaptation methods including CycleGAN and ControlNet and the original HRDA, achieving an mIoU score of 63.0%. However, FDA slightly outperforms SOLO, despite the visually inferior qualitative results of the former in Fig. 2. We hypothesize that this quantitative UDA-based comparison is not suitable to fully demonstrate the superior realism of the nighttime images rendered with



(a) ablation 0 (b) ablation 1 (c) ablation 2 (d) ablation 3 (e) SOLO

Figure 3. **Ablation study of SOLO.** From column (a) - (d) the results of the ablated versions of SOLO i.e. 0, 1, 2, and 3 are presented. SOLO generated nighttime images are displayed in column (e). Notably, every sample (row) is labeled with a number.

Table 2. **Ablation study of SOLO using HRDA framework for the semantic segmentation task.** "geometric": set of geometric components, "lights inst.": set of lights instantiation components, "image post-proc.": set of image post-processing components.

id	geometric	lights inst.	image post-proc.	mIoU
0	×	×	×	52.7 ± 1.6
1	✓	×	×	53.5 ± 1.0
2	×	✓	×	52.4 ± 0.3
3	✓	✓	×	53.6 ± 0.9
4 (SOLO)	✓	✓	✓	55.1 ± 0.4

SOLO. This is because images rendered with SOLO are generally darker than those output by FDA. Consequently, as dark regions in SOLO images appearing as coherent segments may actually include segments from different classes, the inherited daytime annotations may not correspond to discernible segments and thus confuse the model. To further evaluate the realism of of generated images, we use the Kernel Inception Distance (KID) [3] that quantifies the discrepancy between real and generated nighttime images. Notably, according to KID (Table 1) images rendered with SOLO are the most realistic.

4.4. Ablation Study

An ablation study is conducted using four ablated versions of SOLO. For the qualitative evaluation, the generated nighttime images are shown in Fig. 3. To verify the qualitative observations, the HRDA framework for semantic segmentation is employed in Table 2. The ablated versions are formed by 'switching off' component sets. Essential components, such as backprojection, are not ablated. The rest are divided into geometric, light instantiation and image post-processing sets. Disabling the geometric set removes the instance-reference cross-bilateral filter, the normal-guided depth optimization and the mesh post-processing. Similarly, ablating the light instantiation set assumes all light sources are active, sets all light sources color to white, and uses a uniform strength value. Finally, by switching off the image post-processing set, the noise addition and the fog glare effect are disabled.

In Table 2, SOLO outperforms all the ablated versions significantly. Significant difference in mIoU is also observed when either the image post-processing (SOLO → ablation 3) or the geometric (ablation 3 → ablation 2) component sets are ablated. These differences are also evident in the qualitative results of Fig. 3. Specifically, artifacts in the geometry are greatly reduced when the geometric component set is included (ablation 0 → ablation 1) and the image post-processing components result in more realistic renderings, introducing artifacts typical at night time (ablation 3 → SOLO). However, the quantitative results for the lights instantiation ablation are inconclusive. We attribute this finding to the increased brightness of renderings without our light instantiation components, as all light sources are activated for these, as opposed to partial activation with our method (cf. Fig. 3b vs. 3d). This increase leads to more discernible objects, which counteracts the reduced realism when it comes to semantic segmentation performance.

5. Conclusion

We present SOLO, the first monocular, physically-based method for simulating photorealistic nighttime versions of daytime scenes. Our method features several novel contributions, such as a probabilistic light source instantiation module which selectively activates light sources in the scene to achieve more realistic and contextually accurate results. Moreover, we employ a pipeline guided by semantics to fuse geometric representations into a single 3D mesh for usage in forward rendering. Our image post-processing pipeline effectively mimics typical camera artifacts for night time. Our results suggest that SOLO significantly outperforms current state-of-the-art data-driven time-of-day-transfer approaches in the context of day-to-night UDA, highlighting the importance of semantic and physically-based priors in synthesizing photorealistic nighttime images. Finally, we believe that our ACDC Light Sources and Nighttime Illuminants datasets will be valuable resources for the community in working on night time.

Acknowledgment

This work was supported by an ETH Career Seed Award.

References

[1] Cycles: A physically based production renderer developed by the blender project. 6

[2] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[3] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 2, 8

[4] Brent Burley and Walt Disney Animation Studios. Physically-Based Shading at Disney. In *Acm Siggraph*, volume 2012, pages 1–7. vol. 2012, 2012. 5

[5] Anoop Cherian and Alan Sullivan. Sem-gan: Semantically-consistent image-to-image translation. In *2019 ieee winter conference on applications of computer vision (wacv)*, pages 1797–1806. IEEE, 2019. 2

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 5, 6

[7] Dengxin Dai and Luc Van Gool. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *IEEE International Conference on Intelligent Transportation Systems*, 2018. 1

[8] Jian Gao, Chun Gu, Youtian Lin, Hao Zhu, Xun Cao, Li Zhang, and Yao Yao. Relightable 3d gaussian: Real-time point cloud relighting with brdf decomposition and ray tracing. *arXiv preprint arXiv:2311.16043*, 2023. 2

[9] Rui Gong, Qin Wang, Martin Danelljan, Dengxin Dai, and Luc Van Gool. Continuous pseudo-label rectified domain adaptive semantic segmentation with implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[10] Martin Hahner, Christos Sakaridis, Mario Bijelic, Felix Heide, Fisher Yu, Dengxin Dai, and Luc Van Gool. LiDAR snowfall simulation for robust 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[11] Martin Hahner, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Fog simulation on real LiDAR point clouds for 3D object detection in adverse weather. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[12] Shirsendu Sukanta Halder, Jean-Francois Lalonde, and Raoul de Charette. Physics-based rendering for improving robustness to rain. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[13] Eric Heitz, Johannes Hanika, Eugene d'Eon, and Carsten Dachsbacher. Multiple-scattering microfacet bsdfs with the smith model. *ACM Transactions on Graphics (TOG)*, 35(4):1–14, 2016. 6

[14] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning*, 2018. 1

[15] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. HRDA: Context-aware high-resolution domain-adaptive semantic segmentation. In *The European Conference on Computer Vision (ECCV)*, 2022. 2, 6, 7

[16] Lukas Hoyer, Dengxin Dai, Haoran Wang, and Luc Van Gool. MIC: Masked image consistency for context-enhanced domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[17] Sheng-Wei Huang, Che-Tsung Lin, Shu-Ping Chen, Yen-Yi Wu, Po-Hao Hsu, and Shang-Hong Lai. Auggan: Cross domain adaptation with gan-based data augmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 718–731, 2018. 2

[18] Brian Karis and Epic Games. Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice*, 4(3):1, 2013. 5

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[20] Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. In *SIGGRAPH*, 2007. 3

[21] King Man Lam. Metamerism and color constancy. *Ph. D. Thesis, University of Bradford*, 1985. 5

[22] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[23] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse Rendering for Complex Indoor Scenes: Shape, Spatially-Varying Lighting and SVBRDF from a Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2475–2484, 2020. 2, 6

[24] Linjie Lyu, Ayush Tewari, Thomas Leimkühler, Marc Habermann, and Christian Theobalt. Neural radiance transfer fields for relightable novel-view synthesis with global illumination. In *European Conference on Computer Vision*, pages 153–169. Springer, 2022. 2

[25] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[26] Luigi Piccinelli, Christos Sakaridis, and Fisher Yu. idisc: Internal discretization for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21477–21487, 2023. 6

[27] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. *arXiv preprint arXiv:2403.18913*, 2024. 6

[28] Abhijith Punnappurath, Abdullah Abuolaim, Abdelrahman Abdelhamed, Alex Levinshtein, and Michael S Brown. Day-to-Night Image Synthesis for Training Nighttime Neural ISPs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10769–10778, 2022. 2, 5, 6

[29] D Andrew Rowlands. Color conversion matrices in digital cameras: a tutorial. *Optical Engineering*, 59(11):110801, 2020. 5

[30] SAE J3016:APR2021. *Surface Vehicle Recommended Practice – Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. 1

[31] Christos Sakaridis, David Bruggemann, Fisher Yu, and Luc Van Gool. Condition-invariant semantic segmentation. *CoRR*, abs/2305.17349, 2023. 1

[32] Christos Sakaridis, Dengxin Dai, Simon Hecker, and Luc Van Gool. Model Adaptation with Synthetic and Real Data for Semantic Dense Foggy Scene Understanding. In *Proceedings of the european conference on computer vision (ECCV)*, pages 687–704, 2018. 4

[33] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018. 2

[34] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1

[35] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. 1

[36] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Map-guided curriculum domain adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):3139–3153, 2022. 1

[37] Christos Sakaridis, Haoran Wang, Ke Li, René Zurbrügg, Arpit Jadon, Wim Abbeloos, Daniel Olmeda Reino, Luc Van Gool, and Dengxin Dai. ACDC: The adverse conditions dataset with correspondences for robust semantic driving scene perception. *ArXiv e-prints*, 2024. 2, 6

[38] Shen Sang and Manmohan Chandraker. Single-shot neural relighting and svbrdf estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*, pages 85–101. Springer, 2020. 2

[39] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[40] Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David W Jacobs, and Jan Kautz. Neural inverse rendering of an indoor scene from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8598–8607, 2019. 2

[41] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021. 2

[42] Rob Sumner. Processing raw images in matlab. *Department of Electrical Engineering, University of California Sata Cruz*, 2, 2014. 5

[43] Zian Wang, Tianchang Shen, Jun Gao, Shengyu Huang, Jacob Munkberg, Jon Hasselgren, Zan Gojcic, Wenzheng Chen, and Sanja Fidler. Neural fields meet explicit geometric representations for inverse rendering of urban scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8370–8380, 2023. 2

[44] Xin Wei, Guojun Chen, Yue Dong, Stephen Lin, and Xin Tong. Object-based illumination estimation with rendering-aware neural networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 380–396. Springer, 2020. 2

[45] Xinyi Wu, Zhenyao Wu, Hao Guo, Lili Ju, and Song Wang. DANNet: A one-stage domain adaption network for unsupervised nighttime semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 1

[46] Binhui Xie, Shuang Li, Mingjia Li, Chi Harold Liu, Gao Huang, and Guoren Wang. SePiCo: Semantic-guided pixel contrast for domain adaptive semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):9004–9021, 2023. 1

[47] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Neural Information Processing Systems (NeurIPS)*, 2021. 6, 7

[48] Yanchao Yang and Stefano Soatto. FDA: Fourier domain adaptation for semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 6, 7

[49] Ye Yu and William AP Smith. Inverserendernet: Learning single image inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3155–3164, 2019. 2

[50] Oliver Zendel, Markus Murschitz, Martin Humenberger, and Wolfgang Herzner. How good is my test data? Introducing safety analysis for computer vision. *International Journal of Computer Vision*, 125(1):95–109, 2017. 1

[51] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 6, 7

[52] Xianling Zhang, Nathan Tseng, Ameerah Syed, Rohan Bhasin, and Nikita Jaipuria. Simbar: Single image-based scene relighting for effective data augmentation for automated driving vision tasks. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 3718–3728, 2022. 2

[53] Jingsen Zhu, Fujun Luan, Yuchi Huo, Zihao Lin, Zhihua Zhong, Dianbing Xi, Rui Wang, Hujun Bao, Jiaxiang Zheng, and Rui Tang. Learning-based inverse rendering of complex indoor scenes with differentiable monte carlo raytracing. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–8, 2022. 6

[54] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 2, 6, 7