# Sensor Transfer: Learning Optimal Sensor Effect Image Augmentation for Sim-to-Real Domain Adaptation

Alexandra Carlson<sup>1</sup>, Katherine A. Skinner<sup>1</sup>, Ram Vasudevan<sup>2</sup> and Matthew Johnson-Roberson<sup>3</sup>

Abstract- Performance on benchmark datasets has drastically improved with advances in deep learning. Still, crossdataset generalization performance remains relatively low due to the domain shift that can occur between two different datasets. This domain shift is especially exaggerated between synthetic and real datasets. Significant research has been done to reduce this gap, specifically via modeling variation in the spatial layout of a scene, such as occlusions, and scene environmental factors, such as time of day and weather effects. However, few works have addressed modeling the variation in the sensor domain as a means of reducing the synthetic to real domain gap. The camera or sensor used to capture a dataset introduces artifacts into the image data that are unique to the sensor model, suggesting that sensor effects may also contribute to domain shift. To address this, we propose a learned augmentation network composed of physically-based augmentation functions. Our proposed augmentation pipeline transfers specific effects of the sensor model - chromatic aberration, blur, exposure, noise, and color temperature - from a real dataset to a synthetic dataset. We provide experiments that demonstrate that augmenting synthetic training datasets with the proposed learned augmentation framework reduces the domain gap between synthetic and real domains for object detection in urban driving scenes.

#### I. INTRODUCTION

Synthetic datasets are designed to contain numerous spatial and environmental features that are found in the real domain: images captured during different times of day, in various weather conditions, and in structured urban environments. However, in spite of these shared features and high levels of photorealism, images from synthetic datasets are noticeably stylistically distinct from real images. Figure 1 shows a side-by-side comparison of two of widely-used real benchmark vehicle datasets, KITTI [1], [2], Cityscapes [3], and a state-of-the-art synthetic dataset, GTA Sim10k [4], [5]. These differences can be quantified; a performance drop is observed between training and testing deep neural networks (DNNs) between the synthetic and real domains [5]. This suggests that real and synthetic datasets differ in their global pixel statistics. Domain adaptation methods attempt to minimize such dissimilarities between synthetic and real datasets that result from an uneven representation of visual information in one domain compared to the other. Recent domain adaptation research has focused on learning salient

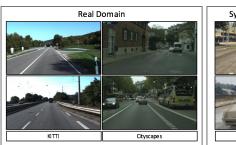




Fig. 1: A comparison of images sampled from the real domain, KITTI Benchmark dataset (shown in the left hand column), images taken from the Cityscapes dataset (shown in the center column), and images from GTA *Sim10k* dataset (shown in the right hand column). Note that each dataset has a distinct visual style, specifically differing color cast, brightness, and blur.

visual features from real data – specifically scene lighting, scene background, weather, and occlusions – using generative adversarial frameworks in an effort to better model the representation of these visual elements in synthetic training sets [6], [7], [8]. However, little work has focused on modelling realistic, physically-based augmentations of synthetic data. Carlson et al. [9] demonstrate that randomizing across the sensor domain significantly improves performance over standard augmentation techniques. The information loss that results from the interaction between the camera model and lighting in the environment is not generally modelled in rendering engines, despite the fact that it can greatly influence the pixel-level artifacts, distortions, and dynamic range, and thus the global visual style induced in each image [10], [11], [12], [13], [14], [15], [16].

In this study, we build upon [9] to work towards closing the gap between real and synthetic data domains. We propose a novel learning framework that performs *sensor transfer* on synthetic data. That is, the network learns to transfer the real sensor effect domain – blur, exposure, noise, color cast, and chromatic aberration – to synthetic images via a generative augmentation network. We demonstrate that augmenting relatively small labeled datasets using *sensor transfer* generates more robust and generalizable training datasets that improve the performance of DNNs for object detection and semantic segmentation tasks in urban driving scenes for both real and synthetic visual domains.

This paper is organized as follows: Section II presents related background work; section III details the proposed

 $<sup>^1</sup>A.$  Carlson and K. A. Skinner are with the Robotics Institute, University of Michigan, Ann Arbor, MI 48109, USA {askc, kskin}@umich.edu

<sup>&</sup>lt;sup>3</sup>R. Vasudevan is with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA ramv@umich.edu

<sup>&</sup>lt;sup>3</sup>M. Johnson-Roberson is with the Department of Naval Architecture and Marine Engineering, University of Michigan, Ann Arbor, MI 48109, USA mattjr@umich.edu

sensor transfer learning framework; section IV describes experiments and discusses results of these experiments and section V concludes the paper. Code will be made publicly available.

#### II. RELATED WORK

Our work focuses on augmenting the training data directly so that it can be applied to any task or input into any deep neural network regardless of the architecture. Zhang et al. 2017 [17] demonstrate that the level of photorealism of the synthetic training data directly impacts the robustness and performance of the deep learning algorithm when tested on real data across a variety of computer vision tasks. However, it remains unclear what features of real data are necessary for this performance gain, or what parts of rendering pipelines should be modified to bridge the synthetic to real domain gap. Much work in the fields of data augmentation and learned rendering pipelines have proposed methods that shed light on this topic, and are summarized below.

#### A. Domain Randomization

Recent work on domain randomization seeks to bridge the sim-to-real domain gap by generating synthetic data that has sufficient random variation over scene factors and rendering parameters such that the real data falls into this range of variation, even if the rendered data does not appear photorealistic. Such scene factors include such as textures, occlusion levels, scene lighting, camera field of view, and uniform noise, and have been applied to vision tasks in complex indoor and outdoor environments [18], [19]. The drawback of these techniques is that they only work if they sample the visual parameters spaces finely enough, and create a large enough dataset from a broad enough range of visual distortions to encompass the variation observed in real data. This can result in intractably large datasets that require significant training time for a deep learning algorithm. While we also aim to achieve robustness via an augmentation framework, we can use smaller datasets to achieve state-of-the-art performance because our method is learning how to augment synthetic data with salient visual information that exists in real data. Note that, because our work focuses on image augmentation outside of the rendering pipeline, it could be used in addition to domain randomization techniques.

## B. Optimizing Augmentation

In contrast to domain randomization, task-dependent techniques have been proposed to achieve more efficient data augmentation by learning the type and number of image augmentations that are important for algorithm performance. State-of-the-art methods [20], [21], [22] in this area treat data augmentation as a form of network regularization, selecting a subset of augmentations that optimize algorithm performance for a given dataset and task as the algorithm is being trained. Unlike these methods, we propose that data augmentation can function as a domain adaptation method. Our learning framework is task-independent, and uses physically based

augmentation techniques to investigate the visual degrees of freedom (defined by physically-based models) necessary for optimizing network performance from the synthetic to real domain.

# C. Image-to-Image Translation for Domain Adaptation

Impressive advances have been made in both paired and unpaired image-to-image translation [23], [24], [25], [26], [27], [28] to bridge a variety of domain gaps, including season-to-season, night-to-day, and sim-to-real. However, image-to-image translation performed between image sets with complex, varied environments often introduces unrealistic distortion artifacts into the underlying structure of the scene. This can yield poor performance for visual tasks such as object detection and semantic segmentation [29]. In contrast, the proposed method does not alter the spatial information in the scene, and instead translates images from one domain to another constrained by physically-based image augmentation.

### D. Learned Rendering Pipelines for Domain Adaptation

Several studies have proposed unsupervised, generative learning frameworks that either take the place of a standard rendering engine [7] or complement the rendering engine via post-processing [30], [31], [32] in order to model relevant visual information directly from real images with no dependency on a specific task framework. Both [7] and [32] are applied to complex outdoor image datasets, but are designed to learn distributions over simpler spatial features in real images, specifically scene geometry. Other methods, such as [30], [31], attempt to learn low-level pixel features. However, they are only applied to image sets that are homogeneously structured and low resolution. This may be due to the sensitivity of training adversarial frameworks. Our work focuses specifically on modeling the camera and image processing pipeline rather than scene elements or environmental factors that are specific to a given task. Our method can be applied to high resolution images of complex scenes.

# E. Impact of Sensor Effects on Deep Learning

Recent work has demonstrated that elements of the image formation and processing pipeline can have a large impact upon learned representation for deep neural networks across a variety of vision tasks [33], [34], [15], [16]. The majority of methods propose learning techniques that remove these effects from images [34]. As many of these sensor effects can lead to loss of information, correcting for them is non-trivial, potentially unstable, and may result in the hallucination of visual structure in the restored image. In contrast, Carlson et al. [9] demonstrate that significant performance boosts can be achieved by augmenting images using physicallybased, sensor effect domain randomization. However, their method requires hand-tuning/evaluation of the visual quality of image augmentation. This human-in-the-loop dependence is inefficient and difficult to scale for large synthetic datasets, and the evaluated visual image quality is subjective. Rather than removing these effects, randomly adding them in, or manually adding them in via human-in-the-loop, our method learns the the style of sensor effects from real data and transfers this *sensor style* to synthetic images to bridge the synthetic-to-real domain gap.

#### III. METHODS

The objective of the sensor transfer network is to learn the the optimal set of augmentations that transfer sensor effects from a real dataset to a synthetic dataset. Our complete Sensor Transfer Network is shown in Figure 2.

A. Sensor Effect Augmentation Pipeline

We adopt the sensor effect augmentation pipeline from [9]. This is the backbone of the Sensor Transfer Network. Refer to [9] for a detailed discussion of each function and its relationship to the image formation process in a camera. We briefly describe each sensor effect augmentation function below for completeness. The sensor effect augmentation pipeline is a composition of chromatic aberration, Gaussian blur, exposure, pixel-sensor noise, and post-processing color balance augmentation functions:

$$I_{aug.} = f_{color}(f_{noise}(f_{exposure}(f_{blur}(f_{chrom.ab.}(I)))))$$
(1)

#### **Chromatic Aberration**

To model lateral chromatic aberration, we apply translations  $(t_x,t_y)$  in 2D pixel space to each of the color channels of an image. To model longitudinal chromatic aberration, we scale the green color channel relative to the red and blue channels of an image by a value S. We combine these parameters into an affine transformation on each pixel in color channel of the image. The augmentation parameters learned for this augmentation function are S, the red channel translations  $R_x$  and  $R_y$ , the green channel translations  $G_x$  and  $G_y$ , and the blue channel translations  $B_x$  and  $B_y$ .

# Blur

We implement out-of-focus blur, which is modeled by convolving the image with a Gaussian filter [35]. We fix the window size of the kernel to 9.0. The augmentation parameter learned for this augmentation function is the standard deviation  $\sigma$  of the kernel.

#### **Exposure**

We implement the exposure density function developed in [36], [37]:

$$I = f(S) = \frac{255}{1 + e^{-A \times S}} \tag{2}$$

where I is image intensity, S models the incoming light intensity, and A is a constant value that describes image contrast. We set A to 0.85. This model is used to re-expose an image as follows:

$$S' = f^{-1}(I) + \Delta S \tag{3}$$

$$I_{exp} = f(S') \tag{4}$$

The augmentation parameters learned for this augmentation function are  $\Delta S$  to model changing exposure, where a positive  $\Delta S$  relates to increasing the exposure, and a negative value indicates decreasing exposure.

#### Noise

We use the Poisson-Gaussian noise model proposed in [12]:

$$I_{noise}(x,y) = I(x,y) + \eta_{poiss}(I(x,y)) + \eta_{qauss}$$
 (5)

where I(x,y) is the ground truth image at pixel location (x,y),  $\eta_{poiss}$  is the signal-dependent poisson noise, and  $\eta_{gauss}$  is the signal-independent gaussian noise. The augmentation parameters learned for this augmentation function are the  $\eta_{poiss}$  and  $\eta_{gauss}$  for each color channel, for a total of six parameters.

#### Post-processing

We model post-processing techniques done by cameras, such as white balancing or gamma transformation, by performing linear translations in LAB color space [38], [39]. The augmentation parameters learned for this augmentation function the are translations in the a (red-green)and b (blue-yellow) channels in normalized LAB space.

# B. Training the Sensor Transfer Network

A high-level overview of a single training iteration for a single sensor effect is given in Figure 3. Each sensor effect augmentation function has its own parameter generator network. The training objective for each of these networks is to learn the distribution over its respective augmentation parameter(s) based upon real data. Each generator network is a two-layer, fully connected neural network. The following steps are required to perform a single training iteration of the Sensor Transfer Network using a single synthetic image. First, a 200 dimensional uniform noise vector,  $\eta$ , is generated and paired with the input synthetic image. The noise vector  $\eta$ is input into each separate generator network. Each generator network consists of two fully connected layers that together project  $\eta$  into its respective sensor effect parameter space. For example, the blur parameter generator will map the  $\eta$ to a value in the  $\sigma$  parameter space. The output sampled parameters, paired with the input synthetic image are then input into the augmentation pipeline, which outputs an augmented synthetic image. This augmented synthetic image is then paired with a real image, both of which are input to the loss function.

We employ a loss function similar to the one used in Johnson et. al [24]. We assume that the layers of the VGG-16 network [40] trained on ImageNet [41] encode relevant style information for salient objects We fix the weights of the pretrained VGG-16 network, and use it to project real and augmented synthetic images into the hidden layer feature spaces. We calculate the style loss, given in Eqn. 6, and use this as the training signal for the parameter generators.

$$L_{style}(y, \hat{y}) = \sum_{j} \|G_{j}^{\theta}(y) - G_{j}^{\theta}(\hat{y})\|_{Frobinius}^{2}$$
 (6)

In the above equation, y is a real image batch,  $\hat{y}$  is an

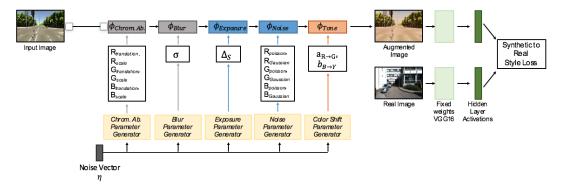


Fig. 2: The schematic of the proposed sensor transfer network structure. The style loss trains the sensor effect parameter generators (represented as the yellow boxes) to select parameters that transform the input synthetic images based upon how the sensor effect augmentation functions alter the style of the real data domain. This effectively transfers 'sensor style' of the target dataset to the source dataset.

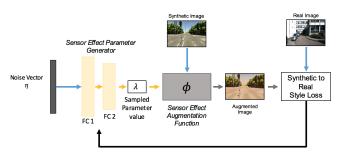


Fig. 3: A detailed schematic of how the training process occurs for a single sensor effect function. A 200 dimensional uniform noise vector (sampled from the range -1 to 1) is generated for a given input synthetic image. The uniform noise vector is input into the fully connected neural network that constitutes the parameter generator, which outputs sampled value(s) for the respective sensor effect augmentation function. The sampled parameter value(s) and the input synthetic image are fed into the sensor effect augmentation function, which outputs an augmented synthetic image. The style loss is calculated between the augmented synthetic image and a real image. This style loss is then backpropagated through the augmentation functions to train the parameter generator to select parameters that reduce the style differences between the real and augmented synthetic images.

augmented synthetic image batch,  $G_j^\theta(y)$  is the Gram matrix of the feature map  $\theta(y)$  of hidden layer j of the pretrained VGG-16 network, and  $G_j^\theta(\hat{y})$  is the corresponding quantity for augmented synthetic images. Through performance-based ablation studies, we found that j=10 gives the best performance, so the style loss is calculated for the first ten layers of VGG-16. Once calculated, the style loss is backpropagated through the sensor effect augmentation functions to train the sensor effect parameter generators. The above process is repeated with images from the synthetic and real datasets until the style loss has converged.

We train the sensor effects generators concurrently to learn the joint probability distribution over the sensor effect parameters. This is done to capture the dependencies that exist between these effects in a real camera. Once training is complete, we can fix the weights of the parameter generators, and use them to sample learned parameters to augment synthetic images. Table I shows the statistics of the learned distributions for sensor effect parameters of different real datasets. See Section IV for analysis and discussion of the learned parameters. Note that style loss was chosen because it is independent of spatial structure of an image. In effect, the augmentation parameter generators learn to sample the distributions of sensor effects in real data as constrained by the style of the real image domain.

# IV. EXPERIMENTS

#### A. Experimental Setup

To verify that the proposed method can transfer the sensor effects of different datasets, we train Sensor Transfer Networks using the following synthetic and real benchmark datasets: GTASim10k [5] is comprised of 10,000 highly photorealistic synthetic images collected from the Grand Theft Auto (GTA) rendering engine. It captures different weather conditions and time of day. The Cityscapes [3] training image set is comprised of 2975 real images collected in over 50 cities across Germany. The KITTI training set [1] is comprised of 7481 real images collected in Karlshue, Germany. We train a Sensor Transfer Network to transfer the sensor style of the KITTI training set to GTASim10k, which is referred to as GTASim10k→KITTI. We also train a Sensor Transfer Network to transfer the sensor style of the Cityscapes training set to GTASim10k, which is referred to as GTASim10k→Cityscapes. To train each Sensor Transfer Network, we use a batch size of 1 and learning rate of  $2e^{-5}$ . We trained each network for 4 epochs. For all experiments, we compare our results to the Sensor Effect Domain Randomization [9] of GTASim10k as a baseline measure to ensure that the transfer of effects is viable over sampling. To generate the Sensor Effect Domain Randomization augmentations, we used the same human-selected parameter ranges as in [9]. To benchmark our method against other, image-based domain



Fig. 4: Qualitative comparison of unaugmented GTASim10k in the first column, Sensor Transfer augmented GTASim10k images in the second column, MUNIT augmented GTASim10k in the third colum, UNIT-augmented GTASim10k in the fourth column, and CycleGAN-augmented GTASim10k in the last column. The first two rows are GTASim10k translated to the KITTI domain, and the second two rows are GTASim10k translated into the Cityscapes domain. Note that, for the Sensor Transfer augmented images, the primary sensor effect transferred in GTASim10k $\rightarrow$ Cityscapes augmentation is decreased exposure, whereas the primary sensor effects transferred in GTASim10k $\rightarrow$ KITTI augmentation is a blueish hue and increased exposure. In comparison, images augmented using the image-to-image translation networks lose a significant amount of spatial information. These methods also cannot handle night time images as well as the proposed method.

adaptation methods, we use the state-of-the-art image-to-image translation methods CycleGAN [23], UNIT [28], and MUNIT [27] as additional baseline measures. Each of the CycleGAN, UNIT, and MUNIT image-to-image translation networks were trained to transfer GTASim10k to Cityscapes, and separately to transfer GTASim10k to KITTI. Each network was trained using either the default hyperparameters provided in the respective paper(s) or until the networks converged.

# B. Evaluation of Learned Sensor Effect Augmentations

Qualitatively, from observing Figure 1, KITTI images feature more pronounced visual distortions due to blur, over-exposure, and a blue color tone. Cityscapes, on the other hand, has a more under-exposed, darker visual style.

Figure 4 shows examples of unaugmented GTASim10k images in comparison to those same images augmented by the proposed Sensor Transfer network and baseline image-to-image translation networks. When compared to Figure 1, it does appear that, for both the sensor transfer of GTASim10k \rightarrow KITTI and GTASim10k \rightarrow Cityscapes, realistic aspects of exposure, noise, and color cast are transferred to GTASim10k. The statistics of the learned parameter values are given in Table I. In general the selected parameter values generate augmented synthetic images with style that matches the real datasets. We hypothesize that the color shift for GTASim10k \rightarrow Cityscapes is not as strong as

GTASim10k $\rightarrow$ KITTI because there is a more even distribution of sky and buildings in Cityscapes, where as KITTI has a significant number of instances of sky. Interestingly, the blur parameter,  $\sigma$ , did not converge and was pushed towards zero for both GTASim10k $\rightarrow$ Cityscapes and GTASim10k $\rightarrow$ KITTI. This suggests that Gaussian blur does not match the blur captured by style of real images. Further research could consider more accurate models of blur, such as motion blur.

# C. Impact of Learned Sensor Transformation on Object Detection for Benchmark datasets

To evaluate if the Sensor Transfer Network is adding in salient visual information for vision tasks in the real image domain, we train an object detection neural network on the unaugmented and augmented synthetic data and evaluate the performance of the object detection network on the real data domains, KITTI and Cityscapes. We chose to use Faster R-CNN as our base network for 2D object detection [42]. Faster R-CNN achieves relatively high performance on the KITTI benchmark dataset. Many state-of-the-art object detection networks that improve upon these results still use Faster R-CNN as their base architecture.

We compare Faster R-CNN networks trained on the proposed method to Faster R-CNN networks trained on unaugmented GTASim10k, GTASim10k augmented using the Sensor Transfer Domain Randomization from Carlson et al., GTASim10k augmented using CycleGAN, GTASim10k

TABLE I: Learned sensor effect parameters for  $GTASim10k \rightarrow Cityscapes$  and  $GTASim10k \rightarrow KITTI$ , and the Sensor Effect Domain Randomization parameters from Carlson et al. [9]. Note that for the Sensor transferred parameters in the first two rows, the mean and standard deviation of each sensor effect parameter value is given in the convention  $\mu \pm \sigma$ . For the Carlson et al. [9] Sensor Effect Domain Randomization parameters, given in the final row, the minimum and maximum of the human selected range is provided. Quantitatively, the  $GTASim10k \rightarrow KITTI$  increases image exposure, adds chromatic aberration, noise, and adds a blue color cast. For  $GTASim10k \rightarrow Cityscapes$ , image exposure is decreased, adds chromatic aberration, a higher level of noise is added, and slight yellow-blue color cast is applied.

Proposed Method	$GTASim10k \rightarrow Cityscapes$	Sensor Effect Parameters

Chrom. Ab.	Blur	Exposure	Noise	Post-processing
$G_{scale}$ : $0.999 \pm 2.398e^-5$	$\sigma$ : $0.718 \pm 1.34e^{-13}$	$\Delta S$ : $-0.273 \pm 0.0249$	$R_{gauss.}$ : $1.0e^-6 \pm 1.382e^-18$	$a:-0.002 \pm 5.239e^{-4}$
$R_{tx}$ : $0.004 \pm 6.221e^{-5}$			$R_{poiss}$ : $1.0e^-6 \pm 1.382e^-18$	$b: -0.0116 \pm 4.727e^{-4}$
$R_{ty}$ : $0.007 \pm 5.511e^{-5}$			$G_{gauss.}$ : 5.41 ± 4.249 $e^-4$	
$G_{tx}$ : $0.005 \pm 1.111e^{-5}$			$G_{poiss}$ : $1.15e^{-2} \pm 7.913e^{-5}$	
$G_{ty}$ : $0.006 \pm 4.718e^{-5}$			$B_{gauss}$ : $1.0e^-6 \pm 1.382e^-18$	
$B_{tx}$ : $0.006 \pm 5.793e^{-5}$			$B_{poiss.}$ : $6.8e^-4 \pm 4.608e^-6$	
$B_{ty}$ : $-5.052 \pm 1.16e^{-4}$				

#### Proposed Method GTASim10k→KITTI Sensor Effect Parameters

Chrom. Ab.	Blur	Exposure	Noise	Post-processing
$G_{scale}$ : $1.001 \pm 6.425e^{-5}$	$\sigma$ : $0.941 \pm 5.173e^{-7}$	$\Delta S$ : 0.0823 $\pm$ 0.003	$R_{gauss.}$ : $9.5e^{-3} \pm 3.713e^{-4}$	$a:-0.0131 \pm 5.426e^{-4}$
$R_{tx}$ : 1.134 $e^-4 \pm 9.416e^-5$			$R_{poiss}$ : $3.07e^{-2} \pm 1.295e^{-3}$	$b: -0.0882 \pm 3.25e^{-3}$
$R_{ty}$ : $-0.0013 \pm 6.874e^{-5}$			$G_{gauss}: 4.5e^{-3} \pm 2.005e^{-4}$	
$G_{tx}$ : $-4.67e^-4 \pm 5.65e^-5$			$G_{poiss.}$ : $2.62e^{-2} \pm 1.111e^{-3}$	
$G_{ty}$ : $-0.0014 \pm 7.228e^{-5}$			$B_{gauss}$ : $2.65e^-2 \pm 1.111e^-3$	
$B_{tx}$ : $-0.003 \pm 1.245e^{-4}$			$B_{poiss}$ : $4.47e^{-2} \pm 1.187e^{-3}$	
$B_{ty}$ : $-5.16e^-5 \pm 1.096e^-4$			-	

Carlson et al. [9] Sensor Effect Domain Randomization Parameters

$G_{scale}$ : 0.998-1.002	$\kappa_{size}$ : 3-11	$\Delta S$ : -0.6-1.2	$R_{gauss.}$ : 0.00-0.05	a: -10.0-10.0
$R_{tx}$ : -0.003-0.003	$\sigma$ : 0.0-3.0		$G_{gauss.}$ : 0.00-0.05	b: -10.0-10.0
$R_{ty}$ : -0.003-0.003			$B_{gauss.}$ : 0.00-0.05	
$G_{tx}$ : -0.003-0.003			$R_{poiss}$ : 0.00-0.05	
$G_{ty}$ : -0.003-0.003			$G_{poiss.}$ : 0.00-0.05	
$B_{tx}$ : -0.003-0.003			$B_{poiss}$ : 0.00-0.05	
$B_{ty}$ : -0.003-0.003			•	

augmented using UNIT, and GTASim10k augmented using MUNIT. To create augmented training datasets, we combine the unaugmented GTASim10k with varying amounts of augmented GTASim10k data. For all datasets, both augmented and unaugmented, we trained each Faster R-CNN network for 10 epochs using two Titan X Pascal GPUs in order to control for potential confounds between performance and training time. We evaluate the Faster R-CNN networks on either the KITTI training dataset or the Cityscapes training dataset depending on the Sensor Transfer Network used for training dataset augmentation. Each dataset is converted into Pascal VOC 2012 format to standardize training and evaluation, and performance values are the VOC AP50 reported for the car class [43].

Table II shows the object detection results for the proposed method in comparison to the image-to-image translation and domain randomization baselines. In general, the addition of sensor effect augmentations has a positive boost on Faster R-CNN performance for training on GTASim10k and testing on Cityscapes. Our proposed method, for both  $GTASim10k \rightarrow Cityscapes$  and  $GTASim10k \rightarrow KITTI$ , achieves the best performance over both the baseline and Sensor Effect Domain Randomization.

To evaluate the impact of Sensor Transfer on the number of synthetic training images required for maximal ob-

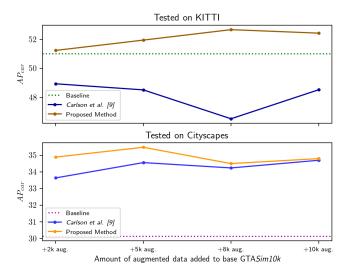


Fig. 5: Results of the learned sensor effect augmentations on Faster R-CNN object detection performance. Note that higher performance can be achieved using smaller synthetic datasets augmented with the proposed method for both KITTI and Cityscapes.

TABLE II: Results of the sensor effects augmentations on Faster R-CNN object detection performance. The percent change for CycleGAN [23], UNIT [28], MUNIT [27], the Carlson et al. [9] and proposed method are calculated relative to the full, unaugmented baseline datasets.

Training Dataset	Tested o	n KITTI
Augmentation Method	$AP_{Car}$	Gain
Baseline	51.01	_
CycleGAN [23]	48.75	↓ -2.25
UNIT [28]	51.21	↑ 0.21
MUNIT [27]	45.50	↓ -5.51
Carlson et al. [9]	48.94	↓ -2.07
Proposed Method	52.67	<b>+1.66</b>

Training Dataset	Tested on Cityscapes
Augmentation Method	$AP_{Car}$ Gain
Baseline	30.13 —
CycleGAN [23]	29.30 ↓ -0.83
UNIT [28]	28.05 ↓ -2.08
MUNIT [27]	26.20 ↓ -3.93
Carlson et al. [9]	34.89 ↑+4.76
Proposed Method	35.48 ↑+5.35

ject detection performance, we trained Faster R-CNNs on datasets comprised of the 10k unagumented GTASim10k images combined with either 2k augmented images, 5k augmented images, 8k augmented images, or 10k augmented images. Figure 5 captures the effect of increasing number of augmentations on Faster R-CNN performance. We see that, when compared to the Sensor Transfer domain randomization method, fewer training images are required when using Sensor Transfer augmentation for both GTASim10k→KITTI and GTASim10k -> Cityscapes. Our results indicate that learning the augmentation parameters allows us to train on significantly smaller datasets without compromising performance. This demonstrates that we are more efficiently modeling salient visual information than domain randomization. Interestingly, the Sensor Effect Domain Randomization method does worse than baseline across all levels of augmentation when tested on KITTI. We expect that this is because humanchosen set of parameter ranges, which are shown in the bottom row of Table I, do not generalize well when adapting GTA Sim10k to KITTI even though they may generate visually realistic images. One reason for this is that the visually realistic parameter ranges selected in [9] where chosen using a GTA dataset of all daytime images, whereas GTASim10k contains an even representation of daytime and nighttime images. This further demonstrates the importance of learning the sensor effect parameter distributions constrained by how they affect the styles of both the real and synthetic image datasets.

#### V. DISCUSSION AND CONCLUSIONS

In general, our results show that the proposed Sensor Transfer Network reduces the synthetic to real domain gap more effectively and more efficiently than domain randomization. Future work includes increasingly the complexity and realism of the Sensor Transfer augmentation pipeline by modeling other, different sensor effects, as well as implementing models that better capture the pixel statistics of real images, such as motion or defocus blur. Other avenues include investigating the impact of task performance and problem space on the sensor effect parameter selection, and evaluating how the proposed method impacts performance for training synthetic datasets rendered with various levels of photorealism.

#### REFERENCES

- [1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Computer Vision and Pattern Recognition (CVPR)*, 2012 IEEE Conference on. IEEE, 2012, pp. 3354–3361.
- [2] J. Fritsch, T. Kuehnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.
- [3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.
- [4] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European Conference on Computer Vision (ECCV)*, ser. LNCS, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9906. Springer International Publishing, 2016, pp. 102–118.
- [5] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?" in *Robotics and Automation (ICRA)*, 2017 IEEE International Conference on. IEEE, 2017, pp. 746–753.
- [6] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," arXiv preprint arXiv:1701.05957, 2017.
- [7] V. Veeravasarapu, C. Rothkopf, and R. Visvanathan, "Adversarially tuned scene generation," arXiv preprint arXiv:1701.00405, 2017.
- [8] C. Sakaridis, D. Dai, S. Hecker, and L. Van Gool, "Model adaptation with synthetic and real data for semantic dense foggy scene understanding," arXiv preprint arXiv:1808.01265, 2018.
- [9] A. Carlson, K. A. Skinner, and M. Johnson-Roberson, "Modeling camera effects to improve deep vision for real and synthetic data," arXiv preprint arXiv:1803.07721, 2018.
- [10] M. D. Grossberg and S. K. Nayar, "Modeling the space of camera response functions," vol. 26, no. 10. IEEE, 2004, pp. 1272–1282.
- [11] F. Couzinie-Devy, J. Sun, K. Alahari, and J. Ponce, "Learning to estimate and remove non-uniform image blur," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1075–1082.
- [12] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical poissonian-gaussian noise modeling and fitting for single-image rawdata," vol. 17, no. 10. IEEE, 2008, pp. 1737–1754.
- [13] A. Andreopoulos and J. K. Tsotsos, "On sensor bias in experimental methods for comparing interest-point, saliency, and recognition algorithms," vol. 34, no. 1. IEEE, 2012, pp. 110–126.
- [14] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 567–576.
- [15] S. Dodge and L. Karam, "Understanding how image quality affects deep neural networks," in *Quality of Multimedia Experience (QoMEX)*, 2016 Eighth International Conference on. IEEE, 2016, pp. 1–6.
- [16] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1422–1430.
- [17] Y. Zhang, S. Song, E. Yumer, M. Savva, J.-Y. Lee, H. Jin, and T. Funkhouser, "Physically-based rendering for indoor scene understanding using convolutional neural networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017, pp. 5057–5065.
- [18] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Intelligent Robots and Systems (IROS)*, 2017 IEEE/RSJ International Conference on. IEEE, 2017, pp. 23–30.

- [19] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," arXiv preprint arXiv:1804.06516, 2018.
- [20] M. Paulin, J. Revaud, Z. Harchaoui, F. Perronnin, and C. Schmid, "Transformation pursuit for image classification," in *Computer Vision and Pattern Recognition (CVPR)*, 2014 IEEE Conference on. IEEE, 2014, pp. 3646–3653.
- [21] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," arXiv preprint arXiv:1805.09501, 2018.
- [22] J. Lemley, S. Bazrafkan, and P. Corcoran, "Smart augmentation learning an optimal data augmentation strategy." *IEEE Access*, vol. 5, pp. 5858–5869, 2017.
- [23] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-toimage translation using cycle-consistent adversarial networks," arXiv preprint arXiv:1703.10593, 2017.
- [24] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for realtime style transfer and super-resolution," in *European Conference on Computer Vision*. Springer, 2016, pp. 694–711.
- [25] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, "Generative visual manipulation on the natural image manifold," in *European Conference on Computer Vision*. Springer, 2016, pp. 597–613.
- [26] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," arXiv preprint, 2017.
- [27] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," arXiv preprint arXiv:1804.04732, 2018.
- [28] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in Neural Information Processing* Systems, 2017, pp. 700–708.
- [29] A. Dundar, M.-Y. Liu, T.-C. Wang, J. Zedlewski, and J. Kautz, "Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation," arXiv preprint arXiv:1807.09384, 2018.
- [30] L. Sixt, B. Wild, and T. Landgraf, "Rendergan: Generating realistic labeled data," arXiv preprint arXiv:1611.01331, 2016.
- [31] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," arXiv preprint arXiv:1612.07828, 2016.
- [32] S. Huang, D. Ramanan, undefined, undefined, undefined, and undefined, "Expecting the unexpected: Training detectors for unusual pedestrians with adversarial imposters," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 00, pp. 4664–4673, 2017.
- [33] C. Kanan and G. W. Cottrell, "Color-to-grayscale: does the method matter in image recognition?" *PloS one*, vol. 7, no. 1, p. e29740, 2012.
- [34] S. Diamond, V. Sitzmann, S. Boyd, G. Wetzstein, and F. Heide, "Dirty pixels: Optimizing image classification architectures for raw sensor data," 2017.
- [35] H. Cheong, E. Chae, E. Lee, G. Jo, and J. Paik, "Fast image restoration for spatially varying defocus blur of imaging sensor," *Sensors*, vol. 15, no. 1, pp. 880–898, 2015.
- [36] S. A. Bhukhanwala and T. V. Ramabadran, "Automated global enhancement of digitized photographs," *IEEE Transactions on Consumer Electronics*, vol. 40, no. 1, pp. 1–10, Feb 1994.
- [37] G. Messina, A. Castorina, S. Battiato, and A. Bosco, "Image quality improvement by adaptive exposure correction techniques," in *Multi-media and Expo*, 2003. ICME '03. Proceedings. 2003 International Conference on, vol. 1, July 2003, pp. I–549–52 vol.1.
- [38] R. S. Hunter, "Accuracy, precision, and stability of new photoelectric color-difference meter," in *Journal of the Optical Society of America*, vol. 38, no. 12, 1948, pp. 1094–1094.
- [39] S. Annadurai, "Fundamentals of digital image processing." Pearson Education India, 2007.
- [40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556
- [41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition*, 2009. CVPR 2009. IEEE Conference on. Ieee, 2009, pp. 248–255.
- [42] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[43] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascalnetwork.org/challenges/VOC/voc2012/workshop/index.html.