Towards Automated Testing and Robustification by Semantic Adversarial Data Generation

Rakshith Shetty[1], Mario Fritz[2], and Bernt Schiele[1]

[1] Max Planck Institute for Informatics, Saarland Informatics Campus
{rshetty,schiele}@mpi-inf.mpg.de
[2] CISPA Helmholtz Center for Information Security
fritz@cispa.saarland

Abstract. Widespread application of computer vision systems in real world tasks is currently hindered by their unexpected behavior on unseen examples. This occurs due to limitations of empirical testing on finite test sets and lack of systematic methods to identify the breaking points of a trained model. In this work we propose semantic adversarial editing, a method to synthesize plausible but difficult data points on which our target model breaks down. We achieve this with a differentiable object synthesizer which can change an object's appearance while retaining its pose. Constrained adversarial optimization of object appearance through this synthesizer produces rare/difficult versions of an object which fool the target object detector. Experiments show that our approach effectively synthesizes difficult test data, dropping the performance of YoloV3 detector by more than 20 mAP points by changing the appearance of a single object and discovering failure modes of the model. The generated semantic adversarial data can also be used to robustify the detector through data augmentation, consistently improving its performance in both standard and out-of-dataset-distribution test sets, across three different datasets.

1 Introduction

Performance evaluation of computer vision systems is predominantly done by empirical evaluation on a fixed test set, often drawn from a similar distribution as the training data. However, due to limited sample size a fixed test set only captures a small portion of errors the model would make on diverse data seen during real-world deployment. This discrepancy manifests as poor out-of-dataset-distribution (OODD) generalization [15, 27, 28], vulnerabilities to input noise [14, 24] and adversarial perturbations [12]. In this work, we propose automated testing through semantic adversarial editing which synthesizes difficult cases, targeted for a particular model, exposing its weaknesses. The error cases synthesized by our model often have atypical appearance and outside the distribution covered by fixed size datasets, however still within the true class boundary to human observers. Apart from its usefulness for testing, our semantic adversarial data can also be used to robustify the target model and improve its performance on OODD data.

To create reliable test data, we need to ensure that the generated sample is consistent with its label. At the same time, the created test data needs to

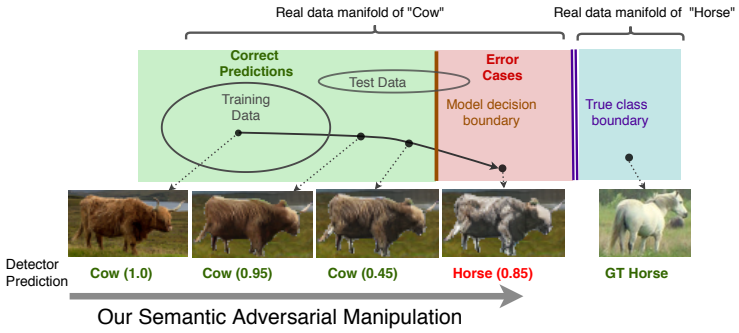

Fig. 1: Standard testing paradigms only covers a small portion of errors models make in the real world due to sample size limitation. We propose semantic adversarial testing to find targeted failure cases through continuous optimization of object appearance to cross the model's decision boundary, while remaining within the true class boundary.

be difficult, ideally capturing the different failure modes of the target model. Simply gathering more data is expensive and inefficient as the process is not targeted to the model. Our approach to meet both these criteria is to start from a real data point, and to make constrained semantic edits through a differentiable synthesizer model. The synthesis process is adversarially optimized to produce semantic changes which fool the target model. By only editing the appearances of individual objects with their pose and the scene held intact, we keep the changes minimal and realistic. An example is seen in Figure 1 where the appearance of "cow" is edited to change the detector prediction to "horse".

Our key insight to constrain the semantic adversarial objects to be label-consistent is by limiting the range of synthesized appearance to be a combination of real ones. We first select a set of guiding templates by sampling instances of the same class from the real data. Then a new appearance is synthesized for the target object optimized to fool the detector, while staying within the convex hull spanned by the appearance of guiding instances. Since changing pose realistically is a much harder task, requiring reasoning over both object and the context, we keep it fixed. Our synthesizer network disentangles the object's pose from its appearance thus allowing editing the appearance without affecting the pose.

Since our semantic adversarial object synthesis process is fully differentiable, we can mine new errors for a target model by directly optimizing the appearance to fool it. We demonstrate this by creating hard test data for the YoloV3 object detector [29]. The same mechanism can also be used to generate hard training data for the detector. The synthesized examples are hard positive examples, often lying close to detectors class boundaries. Our experiments on three dataset, COCO, BDD100k and Pascal, show that using the generated data to fine-tune the detector model improves the model performance and generalization to data distribution shift. To summarize, main contributions of our work are:

– We propose the first method for automatized testing of computer vision models finding new error cases by synthesizing semantic adversarial examples.

– We design an object synthesizer network which disentangles object shape and appearance. This is achieved through a novel binary part segmentation bottleneck which scales better to the diverse object classes.
– We propose a novel mechanism to semantically change the object appearance to fool detectors, while keeping the appearance within the class boundaries as verified by a human study. Experiments show that our semantic adversary editing the appearance of a single object drops the detector performance by 20 mAP points and helps find new vulnerabilities of the model.
– Utility of our generated data is further shown by using it for training the YoloV3 detector. Experiments on three datasets show that the generated data helps improve the detector performance and generalization to OODD data.

2 Related work

Our work connects to four lines of research: robustness testing, semantic adversarial attacks, data augmentation for object detection and generative models. This section will discuss these connections and how our work differs.

Robustness of vision models. CNN based computer vision models generalize poorly when tested on OODD test data. This includes data with various noise [14, 24], translation and scaling [3], rotations [13], out of context objects [31, 34], or test set resampling [27, 28]. With simulated data and 3D rendering, models can also be fooled by unusual object poses [2] and lighting [21]. Difficult natural "adversarial" examples where ImageNet classifiers fail was collected in [15, 32], but through manual and expensive process. In our work we focus on real data and manipulate object appearances to efficiently synthesize error cases for detectors.

Semantic Adversaries. Deep models have also been attacked with semantic adversarial examples. [6, 9] shows we can fool image classifiers by applying the right adversarial translations and rotations to an image. This is generalized to adversarial spatial deformations in [1, 43]. Attempts to semantically change the object appearance have been limited to parametric color distortions [16] and using a generative model for faces [36] and digits [37]. Our work moves beyond the prior research in both scale and scope. We generate semantic adversarial objects by optimizing both appearance and position of the objects and use it to attack detectors on three large datasets with diverse object classes.

Data augmentation for Object detection. A related research area is the data augmentation for object detection which focus on altering individual objects [7, 8, 38, 41, 42]. An early work [42] uses an adversary to partially mask objects to create hard occlusions. Objects are transferred onto new backgrounds for data augmentation with a cut-paste mechanism in [8]. [7] refines this by also heeding to the context for picking a location to paste objects. Yet, this does not take the object pose into account. [38] takes the cut and paste approach further by training a network to predict worst case position, rotation and scale of the added object to fool the detector. Our work, in contrast, resynthesizes the appearance of entire objects to fool the targeted detector, while preserving original context and pose. Thus our synthesis process allows wider range of semantic changes

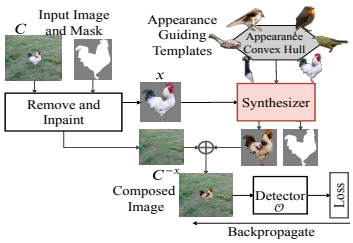

Fig. 2: Our overall pipeline for creating the semantic adversaries

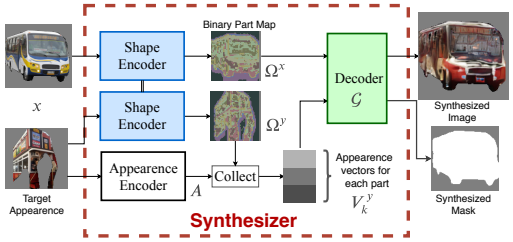

Fig. 3: Synthesizer architecture to generate objects with disentangled appearance and pose latents

compared to occlusions, and better preserves realism and image context compared to cut-and-paste approaches. We compare our approach to a recent work on data augmenting the object detector by switching instances of objects [41]. While [41] circumvents the context issue by switching instances in-place through shape matching, it does not allow generating targeted hard examples.

Unsupervised disentangling of appearence and pose. Our synthesizer architecture is based on unsupervised generative models for disentangling object appearance and pose [17,19,22,35]. Most similar to our design is the model in [22]. In [22], two encoders are used to create latent vectors of pose and appearance, with a Gaussian keypoint bottleneck regulating the pose encoding to carry only spatial information. The key difference in our work is we propose binary segmentation maps as the bottleneck, which scales better to large number of diverse object classes seen in our experiments on COCO dataset.

3 Synthesizing Semantic Adversarial Objects

Our main goal is to efficiently synthesize hard/error cases for an object detector from data manifold. We achieve this goal by starting from a real data point and adversarially editing its appearance through a synthesizer network to fool the target detector. This is a continuous optimization problem efficiently solvable through gradient descent. Additionally, we also need to make sure the synthesized sample is realistic and matches the original label. This is achieved first by only editing appearance of selected objects while retaining its pose, ensuring that the object instance fits well to the image context. Additionally, we constrain the space of appearances allowed during optimization to keep label consistency.

Our solution, shown in Figure 2, consists of two key contributions. First, we build an object synthesizer which disentangles object's pose and appearance, thus allowing us to generate various appearances for an object while keeping its original pose. This is enabled by a binary part segmentation bottleneck, which scales better to diverse object classes, a key requirement to scale to detection datasets like COCO. Second, we propose a novel optimization formulation wherein the latent appearance codes in the synthesizer are constrained to the convex hull of guiding templates. Under this constraint, the appearance is optimized to find the adversarial appearance for an object instance to fool the target detector.

3.1 Synthesizer design

To achieve disentanglement between pose and appearance we propose a modular architecture consisting of an appearance encoder producing latent codes representing the object appearance, a shape encoder producing a binary part segmentation of the object and a decoder which utilizes both the parts and the appearance vectors to synthesize the object. Note that the whole model is learned with only self-supervision, by learning to autoencode objects in the dataset. The overall architecture of synthesizer is shown in Figure 3. While this architecture is inspired by recent works [17, 22], our solution differs in two crucial aspects, the type of bottleneck and the architecture of the decoder. To understand this difference, let us walk through the process of synthesizing an object given an instance x with the target shape and an instance y with target appearance.

Shape Encoder. A representation of the input object shape is first extracted by the shape encoder. This is a CNN with Unet [30] structure which maps the input image into a K×M×N dimensional tensor Z, where K is the number of parts and M,N are the spatial dimensions. Z_{kij} represents the likelihood of the k^{th} part being present at locations ij. To disentangle shape and appearance, we need to restrict Z to only carry information about the spatial layout of the object instance. Prior works [17, 22] do this by approximating Z with 2D Gaussians. While this works for classes like "person" whose parts fit well with gaussian shapes, we find that it does not work well on diverse object classes with complex subparts like "bicycle", "bus", and so on. Instead, we solve this by bottlenecking the information in Z by converting it to a spatial probability distribution and sampling binary masks from it. Specifically, we obtain the part-probability distribution as $P_{kij} = \text{softmax}_k[Z_{ij}]$ and sample binary part maps $\Omega_{kij} = \text{gumbel\_softmax}_k[P_{ij}]$ from it. Here we use gumbel softmax approximation [18, 23] to sample from the multinomial distribution P_{kij} in-order to keep the sampling process differentiable.

Appearance Encoder. Object appearance is encoded with a CNN, which maps the input image to a tensor A of dimensions $D \times M \times N$. This spatial appearance map is reduced to K appearance codes $\mathcal{V} = [V_1 \cdots V_k]$, one for each part, by averaging A over the part activations $V_k = \sum_{ij} P_{kij} Aij$.

Decoder Network. Now using the appearance vector $\mathcal{V}^y$ extracted from image y and the binary part segmentation Ω^x extracted from image x, the decoder network $\mathcal{G}$ synthesizes the desired object and its segmentation mask. The appearance vectors $\mathcal{V}^y_k$ are first projected onto their corresponding binary part activation map to reconstruct the spatial appearance map $\widetilde{A}^y = \mathcal{V}^y \Omega^x$. Our decoder architecture, in contrast to [22], utilizes spatially adaptive normalization layers [25] to input the appearance code at different resolutions to produce the four channel output (image + mask). We find that this helps better preserve the smaller appearance details in generated images as compared to inputting the appearance codes at the first layer. Full network configuration is provided in the supplementary section 1.

Training the Synthesizer. We train the Synthesizer by learning to autoencode objects and to transfer appearance to other instances, similar to prior works [22]. Additionally we use an adversarial discriminator $\mathcal{D}$, to improve the sharpness of

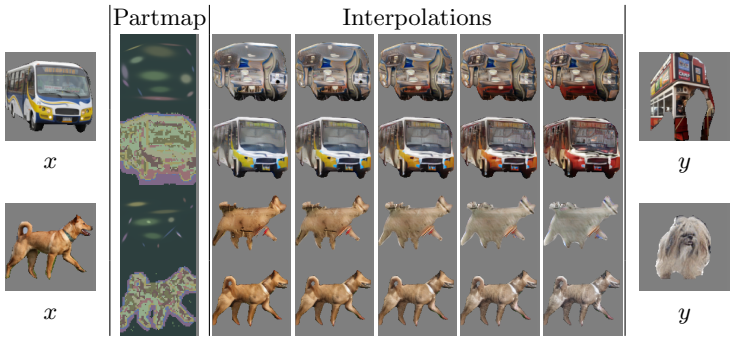

Fig. 4: Appearance interpolations with Our (even rows) and the Gaussian bottleneck model (odd rows). The objects are generated using the shape code from x and by interpolating the appearance vectors from x and y. More examples are in the supplementary.

the generated images. When autoencoding, the model is trained end-to-end with l_1 reconstruction loss for the image and cross-entropy loss for the segmentation mask. Paired training data for learning to transfer appearance is created in two ways. First, we apply simple affine transformations to object instances x to obtain $T(x)$, creating paired data. Now the appearance can be transferred by reconstructing x using shape $\Omega^{T(x)}$ and appearance $\mathcal{V}^x$ encodings and vice-versa. Secondly, the model is trained to transfer appearance to a random instance y of the same class by using the discriminator real/fake loss and cyclic reconstruction loss [45]. Precisely, given shape code Ω^y and $\mathcal{V}^x$, we generate a hybrid object xy and use the discriminator $\mathcal{D}$ to evaluate realism and provide a training signal. We also re-encode xy to obtain appearance code $\mathcal{V}^{xy}$, and use it to reconstruct the original image as $\widetilde{x} = \mathcal{G}\left(\Omega^x, \mathcal{V}^{xy}\right)$. Apart from the reconstruction losses, we also impose additional constraints on the appearance and shape latent codes to provide intermediate supervision. For example, $\Omega^{T(x)}$ should be same as $T(\Omega^x)$ since an affine transformed input image should lead to an affine transformed part-map. Equations for these training losses are given below.

$$L_{\mathrm{r}} = |x - \mathcal{G}(\Omega^x, \mathcal{V}^x)| + |T(x) - \mathcal{G}(\Omega^{T(x)}, \mathcal{V}^x)| + |x - \mathcal{G}(\Omega^x, V^{xy}) \tag{1}$$

$$L_{\mathrm{d}} = \mathcal{D}(\mathcal{G}(\Omega^x, \mathcal{V}^x)) + \mathcal{D}(\mathcal{G}(\Omega^{T(x)}, \mathcal{V}^x)) + \mathcal{D}(\mathcal{G}(\Omega^y, \mathcal{V}^x)) \tag{2}$$

$$L_{\mathrm{a}} = \|\mathcal{V}^x - \mathcal{V}^{T(x)}\| + \|\mathcal{V}^x - \mathcal{V}^{xy}\| \tag{3}$$

$$L_{\mathrm{p}} = -P^{T(x)} \log(T(P^x)) \tag{4}$$

Figure 4 compares the appearance transfer produced by our model trained on COCO dataset and a baseline model with identical structure, except using 2D Gaussians to bottleneck the shape encoding. We see big difference in quality of the generated images especially for objects like bus and dog. This performance gap can be understood by looking at the part representations extracted using the two methods also shown in Figure 4. We see that while Gaussian part maps are very crude approximations, our binary part maps captures detailed shape information, enabling better reconstruction and interpolation of appearance.

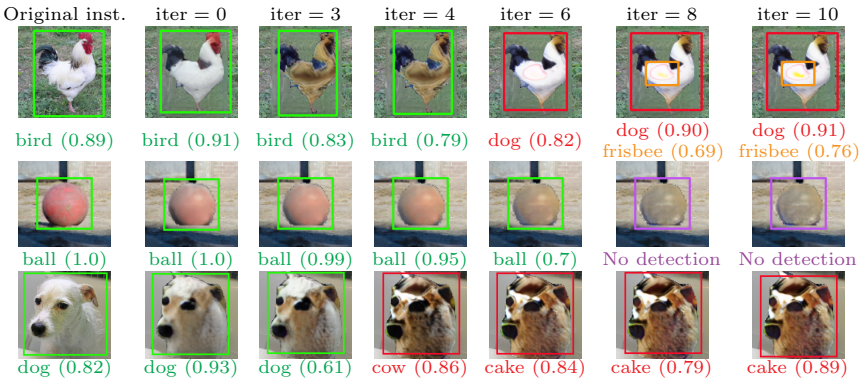

Fig. 5: Intermediate steps when optimizing the appearance to fool the detector.

3.2 Synthesizing Semantic Adversaries

Now that we have a synthesizer which can effectively change appearance of a target object x using an appearance guiding template, let us leverage it to produce semantic adversaries to fool an object detector. We start by extracting the shape (Ω^x) and appearance ($\mathcal{V}^x$) representations for the target instance x occurring in image C, which we wish to edit to fool the detector $\mathcal{O}$. Instance x is removed from C using the ground-truth box and an object removal in-painter from [33] to obtain canvas image C^{-x}. A new version of object x is synthesized as $\mathcal{G}(\Omega^x, \mathcal{V}^x)$ and is pasted in place of the original to get the composed image. We denote this as $C^{-x} + \mathcal{G}(\Omega^x, \mathcal{V}^x)$. This process is illustrated in Figure 2.

A simple way to fool the detector would be to adversarially optimize the appearance vector $\mathcal{V}^x$ until the object detector fails on the generated image $\mathcal{G}(\Omega^x, \mathcal{V}^x)$. However, in unconstrained optimization the appearance vectors often move into areas where synthesizer produces unrealistic images, which also fools the detector. We overcome this with a novel scheme which keeps the adversarially optimized $\mathcal{V}^x$ from going far from the synthesizer's input distribution. We first sample a set $I = \{i_1, \cdots i_n\}$ of n guiding templates belonging to the same class and extract appearance codes for each of them $\mathcal{V}^I = \{\mathcal{V}_k^{i_1}, \cdots \mathcal{V}_k^{i_n}\}$. Now the appearance vector for the generated object is optimized to fool the detector while constraining it to remain within the convex hull spanned by $\mathcal{V}^I$.

$$\mathcal{V}_k^{adv} = \left\{ \sum_{j=1}^{n} \alpha_1^j V_1^{i_j}, \cdots \sum_{j=1}^{n} \alpha_k^j V_k^{i_j} \right\} \tag{5}$$

$$\max_{(a_1^1, \cdots a_k^n)} \mathcal{L}_{det} \left[\mathcal{O} \left(C^{-x} + G \left(\Omega^x, \mathcal{V}_k^{adv} \right) \right) \right] \tag{6}$$

Here $\alpha_k^j = \text{softmax}_n(a_k^j)$, with $\{a_k^1 \cdots a_k^n\}$ being the interpolation co-efficients for part k and $\mathcal{L}_{det}$ is the detector loss function which we maximize. There are total of $n \times k$ interpolation coefficients which are optimized to find the adversary. Having independent part coefficients allows mixing and matching appearances

from different templates for each part, and thus allowing richer appearances space to be explored through optimization. Further, since we only manipulate the latent appearance codes, the adversary cannot directly manipulate pixels to produce noisy patterns to fool the detector, but must instead rely on semantic changes. Detector loss is usually a sum of classification, objectness and box regression losses. We discard the box regression losses, as they are not directly affected by appearance and often leads to unstable behavior in optimization. Hence the detector loss becomes $\mathcal{L}_{det} = \lambda L_{obj} + (1 - \lambda) L_{cls}$, where $\lambda \in [0, 1]$ is a co-efficient controlling how much the adversary focusses on causing missed detection versus misclassification. Spatial perturbations like position or scale of the object can be easily incorporated into our formulation by inserting a parametrized affine transformation matrix before pasting the object onto canvas image, allowing position and appearance to be jointly optimized to fool the detector.

Figure 5 depicts the adversarial appearance optimization steps to fool the YoloV3 detector. First row shows the synthesized bird changing from a reconstruction in the zeroth step to a different color by the fourth step, causing detector confidence to drop. More optimization leads to a bigger failure in the detector where the brown head and the yellow circle in the body of the synthesized bird causes the detector to see the object as a "dog" and a "frisbee". The second row shows a case where the appearance of the "ball" is slowly changed to camouflage with the background and cause a missed detection. These examples show that our method makes large semantic changes to the object appearance which fools the detector while looking plausible to human eye (empirically verified in Section 4.2).

4 Experiments and Results

We evaluate our semantic adversary for two applications, as a diagnostic tool to find failure modes in the detector and as a hard data generation mechanism to improve the performance of these detectors. We measure the effectiveness of the semantic adversary in terms of the detector performance on generated adversarial test set. We verify label consistency of the semantic adversary by a human study where observers verify if the original class is preserved after adversarial editing. We also qualitatively examine the synthesized error cases and find different mechanisms which cause detector failures. Data augmentation experiments are run on three different datasets, COCO, VOC and BDD100k, and we measure the benefit of the generated adversarial data for improving model performance on both standard test, as well as generalization to out-of-dataset-distribution. First, we describe the experimental setup and datasets, followed by the analysis on effectiveness of the semantic adversary for diagnostics and data augmentation.

4.1 Setup and Datasets

We conduct our data augmentation experiments on three datasets – COCO [20], PascalVOC [10](VOC) and BDD100k [44]. COCO and VOC contains both indoor and outdoor images with common objects like person, car, table etc. While

COCO has 120k training images with 80 classes, VOC is smaller with 20 classes and 14k training data (combining 2007+2012 splits). BDD100k is a large scale driving dataset with 100k street scenes captured from a car driving around major US cities, with annotation of objects like person, car, traffic light and so on. The object synthesizer and removal inpainter are both trained on the COCO dataset, due to availability of instance segmentation masks needed to extract the object patches. Since all the classes in VOC and 9/10 classes in BDD100k are part of COCO (except "rider" class), COCO trained model can be used to synthesize adversarial objects on these datasets. The synthesizer operates at 128×128 resolution. The generated objects are scaled to match the target box.

We use the YoloV3 [29] model as the target detector, as it is a popular single staged detector with fast runtime, making adversarial attack experiments run quicker. We train our baseline model from scratch using the implementation available in [39], using all the standard data augmentation methods including color jittering and rotation. However, to keep the synthesis single resolution, all our detector models are trained on single fixed resolution (416×416 on COCO and VOC, 704×1248 on BDD100k) as opposed to multi-scale training used in YoloV3, yielding a lower baseline performance. All the improvement reported from training on our synthesized data is in-complimentary to the standard augmentations. The evaluation is also performed at these fixed resolutions. The models on BDD100k and VOC are trained after initializing from a trained COCO model. This ensures that these models have already been exposed to the instances from the COCO dataset. When training on synthetic data, we start from the pre-trained model and fine-tune the last two layers in case of COCO and BDD100k and last three layers in case of Pascal. For fair comparison we also further fine-tune the pre-trained model using the exact same configuration, but only with real data to obtain the *Base-FT* model in all three datasets.

Apart from evaluating on i.i.d test sets, we also measure the generalization to OODD data. This tests our hypothesis that semantic adversarial data improves the model robustness to OODD samples, since our adversarial data often contains atypical objects, from the tail of appearance distribution. To do this, we test the COCO trained model on the UnRel [26] and VOC test sets. The models are tested on the overlapping 29 classes in UnRel and all 20 classes in VOC. UnRel data contains objects in unusual relationships and contexts and will measure if the model generalizes to rare cases. Similarly VOC trained models are also tested on UnRel, for the overlapping 14 classes. The BDD100k models are tested on D2-City [4], with driving images from Chinese cities.

4.2 Semantic Adversary for Automated Testing

To quantify the effectiveness of the semantic adversary, we create adversarial test sets using the COCO training images by optimizing the appearance of selected objects in each image to fool the detector. Objects are selected at random as long as they are not too small/large (≥ 32 pixels and $\leq 30\%$ of the image area). We do this with three variants of our approach. First only optimizes the appearance of one object instance. The second variant optimizes both the position and the

Optimize	n_obj	mAP ↓	Success rate by instance type ↑		
			Edited	Co-occuring	Combined
Real Data	0	81.2	-	-	-
Appear	1	62.4	74.99	58.62	61.10
Pos + Appear	1	59.5	**77.69**	59.30	62.13
Pos + Appear	2	**46.5**	77.10	**61.54**	**65.82**

Table 1: Overall and instance-level detector performance under semantic advesarial editing. *Co-occuring* refers to the other untouched objects in the image.

Instance	Label correctness
Real	99%
Random label	11%
SemAdv (appearance)	93%

Table 2: Human study results on the label correctness of semantic adversarial editing.

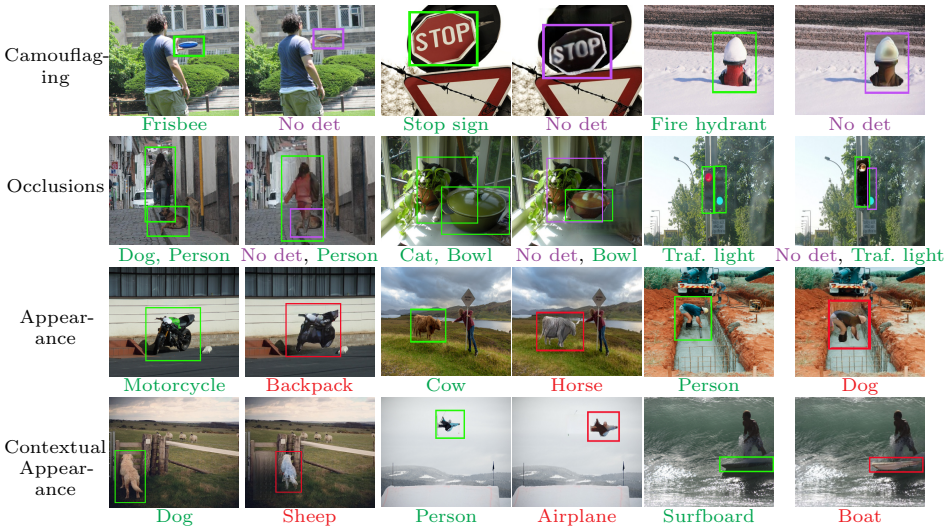

Fig. 6: Qualitative examples of the failure cases discovered by our semantic adversary. **Green** boxes are correct detections, **purple** boxes indicate missed detections and **red** boxes show the misclassified objects. Only relavant detections are marked.

appearance of the same object. In the third variant two random objects are chosen from each image and their position and appearance are adversarially optimized. Each of these test sets contain 37k images. Object detector is run on these three sets and performance is measured using mean average precision (mAP@0.5)

Quantitative Analysis. The results are reported in Table 1. We see that all the semantic adversaries drop the performance of the model significantly, with mAP dropping from 81.2 on the corresponding real data to 62.4 with just optimizing the appearance. Optimizing the position and scale of the object along with appearance further degrades the detector performance, with mAP dropping to 59.5. When we adversarially modify two objects jointly, the detector performance drops again to 46.5 mAP, making it a 57% drop in detector performance. To understand this performance drop, we look at the effect on the detector's confidence for each object instance. We consider it a success if the detector's confidence drops after

adding the semantic adversary. Table 1 presents the success rate on the edited as well as untouched objects in the same image. Firstly, we see that all three strategies drop the detector's confidence on more than 74% of the semantically edited instances. Interestingly, about 60% of the untouched co-occurring instances are also negatively affected. This is often due to the contextual changes caused by the misclassification of the edited instances or minor occlusions produced by the edited instance. We note again that our semantic adversarial attacks are efficient, performed in just 10 steps of gradient descent. To put this in context, adversarial color jittering [16] takes about 200 trials to attack (success in 50% of cases) a simpler classification model on a smaller CIFAR-10 dataset.

We also compare the effectiveness of our semantic adversary to a standard L_∞ norm adversarial attack. For fair comparison we also restrict the L_∞ attacker to change pixels within the bounding box of a single object. The experiments show that our single object semantic adversarial attack (mAP=59.5) is roughly equivalent in strength to a L_∞ norm attack with $\epsilon = 8/255$ (mAP=58.7). Full results and details are presented in the supplementary material.

Human Study. A natural question at this point is if the semantic adversarial samples are within the true class boundary. To answer this we turn to human observers. We conduct a study where a human judge is presented with an image and asked if the object highlighted with the box belongs to the specified class. If they consider the label correct, they are asked to also rate how typical the object appearance is from 1 to 5, with 1 corresponding to very unusual and 5 corresponding to very typical appearance. The study is conducted on a mix of 250 real and semantically edited instances each, with each instance rated by three independent observers. We also introduce additional 10% samples where labels are shuffled. This is done in-order to verify the work of the human annotators. More details including exact instructions and interface is presented in the supplementary section 3. Table 2 shows the label correctness results as judged by majority vote of three humans. As expected the label correctness is very high on real instances and is very low on label-shuffled instances. We also see that in 93% of cases, humans agree that semantic adversary preserves the label of the object instance. Performance drop on semantic adversary is small for human observers compared to the significant drop by object detectors seen before. The typicality rating provided by the human judges on real and semantically edited instances shown in Figure 7 helps understand this gap. While most of the real samples have a typicality rating of 4 or 5 (very typical), semantic adversary have lower rating between 2-3. These results show that the semantic adversary generates atypical examples which are still correctly detectable by humans, but are hard for our detectors. This is further supported by lower performance of the detector on less typical real samples (Accuracy>70% on typicality rating=5 and accuracy 35-40% on typicality rating between 1-3). Further details are in supplementary section 3.

Qualitative Analysis. Examining the cases where the semantic adversary fools the detector reveals that the adversary causes missed detections and misclassification through four main mechanisms listed below and illustrated in Figure 6. All examples are from the strategy with editing single object and position.

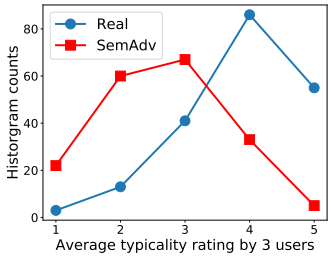

Fig. 7: Comparing the typicality rating between real data and semantic adversary.

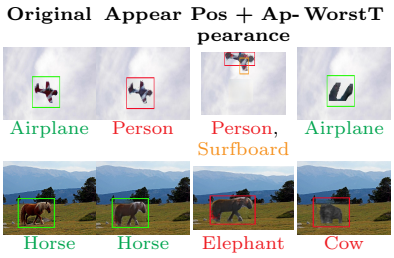

Fig. 8: Comparing various adversarial strategies and the worst-case template baseline.

– **Camouflaging** - Semantic adversary often causes missed detections by changing the appearance of the object to blend with the background. First row of Figure 6 illustrates a frisbee, a stop-sign and a hydrant being camouflaged.
– **Occlusion** - Second row shows cases where the semantic adversary causes missed detections by moving to partially occlude some co-occurring objects.
– **Appearance** - In many instances, the appearance of the object is altered to include small visual features which trigger misclassification by the detector. These can be seen in the third row in Figure 6. We see that "cow" is changed to "horse" based on color change and person is misclassified as a dog due to a small change in hue. These cases indicate that detectors often rely on false correlations of low-level textures or colors to certain classes, and often fail when these textures are altered, as also shown for classifiers in recent work [11].
– **Contextual Appearance** - Last row shows examples where with a change in an object appearance, the contextual evidence overrides the visual features, causing misclassification. Eg. in the first image, a dog changed to white color is misclassified as a sheep as there are other sheep present nearby. Similarly, a falling person is mistaken as an airplane, and a surfboard as a boat.

We note that despite a few generation artifacts, with the COCO dataset being hard for current GAN models, these samples look plausible to human eye and we would not make the same predictions as the detector. This makes it a useful tool to explore the breaking points of a trained detector.

4.3 Semantic Adversary for Data Augmentation

Apart from being a useful diagnostic tool, semantic adversaries can be used to generate training data. By targeting the detector, we can create tailored hard positives for the model, and thus get the most benefit when added to the model training set. We generate the training data with a similar process as in the previous section: first selecting an eligible object from each training image, adversarially optimizing its appearance and adding it back to the training set. The model is then fine-tuned with a combination of the original and the synthesized adversarial data for 50 epochs, and performance on the standard test sets and OODD data is measured. We now present this data augmentation results on the three datasets.

COCO dataset. Table 3 shows the data augmentation results on the COCO dataset. Comparing the *Baseline* and *Base+FT* models we see that the further fine-tuning the last two layers of the model improves the performance a bit on COCO and VOC test sets, while reducing a bit on UnRel (39.0 vs 38.8). Comparing this with the basic semantic adversary augmented model *SA-Rand-App*, which only edits object appearance, we see a bigger improvement on all three test sets. Table 3 also shows *Base+FreeAdv*, an adversarial baseline which allows for free manipulation of appearance vector under l_∞ constraint, without the convex hull constraint. Its poor performance compared to *SA-Rand-App* shows that the unconstrained attack does not work well as often the adversarial sample looks unrealistic. *SA-Rand-App* model generates the semantic adversaries using randomly sampled instances for guidance. We can further target the model weaknesses by sampling the templates from hard instances for the detector, i.e. setting the probability of picking an instance inversely proportional to the detectors confidence on it. The model trained this way, *SA-App*, further improves the performance a bit on COCO and significantly on UnRel (39.6 vs 39.2). Moreover, the *SA* model which jointly optimizes position and appearance gets even better results, improving over *SA-App* in COCO and VOC test sets. Now, we compare our approach to a simpler baseline, where an object is replaced with the template which increases the detector loss the most. While this often fools the detector, it also places instances which do not fit with the image context, as seen in the examples in Figure 8. Thus, the *Base-WorstT* model using this data in training performs worse than *SA-App* on all test sets.

We can increase the benefit of semantic adversaries by editing more objects in the image, creating harder data. The *SA#2* model which edits appearance and position of two objects improves on COCO(+0.3 mAP) and UnRel(+0.4 mAP) test sets compared to *SA* editing single objects. Since our adversarial data is adaptive to the model, we can continue generating harder examples attacking the newly trained model. *SA#2x2* does this, further training the *SA#2* model using the adversarial data generated by attacking *SA#2*. This second iteration helps and *SA#2x2* still improves. By repeating this four times, we get the *SA#2x4* model which outperforms the baseline on all three test sets, COCO(+1.0 mAP), VOC(+1.3 mAP) and UnRel(+1.8 mAP). The gain is larger on OODD test sets, VOC and UnRel, indicating that training with semantic adversary improves the robustness of the model to input distribution changes. We also compare our approach to recent data augmentation approaches PSIS [41] and AutoAug [5]. For PSIS, we use the data provided by the authors [40] to fine-tune our baseline model same as before. While the PSIS data improves over the baseline, it falls short compared to our *SA#2x4* model in all test sets. Table 3 also shows that our approach is complimentary to AutoAug [5], which applies augmentation policies on the entire image. While auto-augment improves the baseline performance, our *SA#2* model improves even more when combined with AutoAug.

PascalVOC Dataset. Results in Table 4 for data augmentation on VOC dataset show that, semantic adversarial data improves performance here as well. The *SAx5* model, which edits appearance and position of a single object, is better than the

Model	obj	COCO	VOC	UnRel
Baseline	-	46.1	66.4	39.0
Base+FT	-	46.2	66.9	38.8
Base+FreeAdv	1	45.8	65.3	37.9
Base+WorstT	1	46.2	66.8	39.2
SA-Rand-App	1	46.5	67.1	39.2
SA-App	1	46.6	67.0	39.6
SA	1	46.7	67.3	39.4
SA#2	2	46.9	67.4	39.8
SA#2 x2	2	47.0	67.4	40.4
SA#2 x4	2	**47.1**	**67.7**	**40.8**
Base+AutoAug [5]	-	47.0	67.6	40.4
SA#2+AutoAug [5]	2	47.8	68.1	41.5
PSIS [41]	-	46.7	67.5	39.8

Table 3: Data augmentation results on COCO dataset. Metric used is mAP@0.5

Model	obj	VOC	UnRel
Base+FT	0	74.0	42.9
Base+WorstT	1	73.7	43.4
SA x5	1	**75.0**	**45.0**
SA#2 x5	2	74.0	44.3

Table 4: Data augmentation results on VOC using semantic adversary.

Model	λ	BDD	D2City
Base+FT	-	50.7	34.7
SA-App	0.5	50.8	34.6
SA-App	1.0	**51.4**	**35.1**
SA	1.0	51.2	35.0

Table 5: Data augmentation results on BDD100k dataset.

baseline on both VOC(+1 mAP) and the UnRel(+2.1mAP) test sets, again with bigger gains on the OODD data. *SA#2x5* which creates two adversarial objects underperforms *SAx5*, since VOC images often have only a single object, which causes the *SA#2x5* to add too many out-of-context objects in its generation.

BDD100k Dataset. On BDD100k (see Table 5), we found that adversary often caused drastic appearance changes when fooling the classifier. Since BDD100k has only 10 classes, the class boundaries are well separated and fooling the classifier needs large unrealistic appearance changes. Instead, optimizing to only reduce the objectness score (setting $\lambda = 1$) leads to more realistic synthesis. This is seen when comparing *SA-App* models with $\lambda = 0.5$ and $\lambda = 1.0$. The model with $\lambda = 1.0$ performs much better on both the BDD and the OODD D2-city test sets, while also improving over the fine-tuned baseline. Additionally, we see in Table 5 that the *SA-App* performs better than *SA* which optimizes position, showing that it is better to edit objects in-place in structured scenes in BDD.

5 Conclusions

We present a method for automatic test case generation through semantic adversarial optimization of object appearances. Our approach can synthesize new OODD hard examples which cause failures in the target detector, while remaining realistic to human eye. Analysis of the synthesized data shows the different failure modes discovered by the process includes camouflaging, occlusions and appearance changes. Our adversarial data is also useful for data augmentation, consistently improving the detector on standard and OODD test sets, in three datasets. We hope that our work will facilitate future approaches to test models beyond finite datasets and hence develop more reliable performance metrics.

References

1. Alaifari, R., Alberti, G.S., Gauksson, T.: Adef: an iterative algorithm to construct adversarial deformations. Proceedings of the International Conference on Learning Representations (ICLR) (2019) 3
2. Alcorn, M.A., Li, Q., Gong, Z., Wang, C., Mai, L., Ku, W.S., Nguyen, A.: Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019) 3
3. Azulay, A., Weiss, Y.: Why do deep convolutional networks generalize so poorly to small image transformations? Journal of Machine Learning Research (JMLR) **20**(184), 1–25 (2019) 3
4. Che, Z., Li, G., Li, T., Jiang, B., Shi, X., Zhang, X., Lu, Y., Wu, G., Liu, Y., Ye, J.: D2-city: A large-scale dashcam video dataset of diverse traffic scenarios. arXiv preprint arXiv:1904.01975 (2019) 9
5. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 113–123 (2019) 13, 14
6. Dumont, B., Maggio, S., Montalvo, P.: Robustness of rotation-equivariant networks to adversarial perturbations. arXiv preprint arXiv:1802.06627 (2018) 3
7. Dvornik, N., Mairal, J., Schmid, C.: Modeling visual context is key to augmenting object detection datasets. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. pp. 375–391. Springer International Publishing, Cham (2018) 3
8. Dwibedi, D., Misra, I., Hebert, M.: Cut, paste and learn: Surprisingly easy synthesis for instance detection. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 1301–1310 (2017) 3
9. Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., Madry, A.: Exploring the landscape of spatial robustness. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the International Conference on Machine Learning (ICML). vol. 97, pp. 1802–1811. PMLR, Long Beach, California, USA (09–15 Jun 2019) 3
10. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html 8
11. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In: Proceedings of the International Conference on Learning Representations (ICLR) (2019) 12
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (NeurIPS) (2014) 1
13. Hamdi, A., Ghanem, B.: Towards analyzing semantic robustness of deep neural networks. Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops) (2019) 3
14. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: Proceedings of the International Conference on Learning Representations (ICLR) (2019) 1, 3
15. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. arXiv preprint arXiv:1907.07174 (2019) 1, 3

16. Hosseini, H., Poovendran, R.: Semantic adversarial examples. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops). pp. 1614–1619 (2018) 3, 11
17. Jakab, T., Gupta, A., Bilen, H., Vedaldi, A.: Unsupervised learning of object landmarks through conditional image generation. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems (NeurIPS). pp. 4016–4027. Curran Associates, Inc. (2018) 4, 5
18. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. Proceedings of the International Conference on Learning Representations (ICLR) (2016) 5
19. Li, Y.J., Lin, C.S., Lin, Y.B., Wang, Y.C.F.: Cross-dataset person re-identification via unsupervised pose disentanglement and adaptation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 7919–7929 (2019) 4
20. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 740–755. Springer International Publishing, Cham (2014) 8
21. Liu, H.T.D., Tao, M., Li, C.L., Nowrouzezahrai, D., Jacobson, A.: Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer. In: Proceedings of the International Conference on Learning Representations (ICLR) (2019) 3
22. Lorenz, D., Bereska, L., Milbich, T., Ommer, B.: Unsupervised part-based disentangling of object shape and appearance. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10947–10956 (2019) 4, 5
23. Maddison, C.J., Mnih, A., Teh, Y.W.: The concrete distribution: A continuous relaxation of discrete random variables. Proceedings of the International Conference on Learning Representations (ICLR) (2016) 5
24. Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A.S., Bethge, M., Brendel, W.: Benchmarking robustness in object detection: Autonomous driving when winter is coming. arXiv preprint arXiv:1907.07484 (2019) 1, 3
25. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 5
26. Peyre, J., Laptev, I., Schmid, C., Sivic, J.: Weakly-supervised learning of visual relations. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017) 9
27. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do cifar-10 classifiers generalize to cifar-10? arXiv preprint arXiv:1806.00451 (2018) 1, 3
28. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do ImageNet classifiers generalize to ImageNet? In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the International Conference on Machine Learning (ICML). vol. 97, pp. 5389–5400. PMLR, Long Beach, California, USA (09–15 Jun 2019) 1, 3
29. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018) 2, 9
30. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015) 5
31. Rosenfeld, A., Zemel, R., Tsotsos, J.K.: The elephant in the room. arXiv preprint arXiv:1808.03305 (2018) 3

32. Shankar, V., Dave, A., Roelofs, R., Ramanan, D., Recht, B., Schmidt, L.: A systematic framework for natural perturbations from videos. Proceedings of the International Conference on Machine Learning Workshops (ICML Workshop) (2019) 3
33. Shetty, R., Fritz, M., Schiele, B.: Adversarial scene editing: Automatic object removal from weak supervision. In: Advances in Neural Information Processing Systems (NeurIPS) (2018) 7
34. Shetty, R., Schiele, B., Fritz, M.: Not using the car to see the sidewalk–quantifying and controlling the effects of context in classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8218–8226 (2019) 3
35. Siarohin, A., Lathuilire, S., Tulyakov, S., Ricci, E., Sebe, N.: Animating arbitrary objects via deep motion transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2372–2381 (2019) 4
36. Song, Y., Shu, R., Kushman, N., Ermon, S.: Constructing unrestricted adversarial examples with generative models. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems (NeurIPS). pp. 8312–8323. Curran Associates, Inc. (2018) 3
37. Stutz, D., Hein, M., Schiele, B.: Disentangling adversarial robustness and generalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6976–6987 (2019) 3
38. Tripathi, S., Chandra, S., Agrawal, A., Tyagi, A., Rehg, J.M., Chari, V.: Learning to generate synthetic data via compositing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 461–470 (2019) 3
39. Ultralytics: Pytorch implementation of YoloV3. https://github.com/ultralytics/yolov3 (2019), [Online; accessed 11-Nov-2019] 9
40. Wang, H.: Implentation of data augmentation for object detection via progressive and selective instance-switching. https://github.com/Hwang64/PSIS (2019), [Online; accessed 11-Nov-2019] 13
41. Wang, H., Wang, Q., Yang, F., Zhang, W., Zuo, W.: Data augmentation for object detection via progressive and selective instance-switching. arXiv preprint arXiv:1906.00358 (2019) 3, 4, 13, 14
42. Wang, X., Shrivastava, A., Gupta, A.: A-fast-rcnn: Hard positive generation via adversary for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2606–2615 (2017) 3
43. Xiao, C., Zhu, J.Y., Li, B., He, W., Liu, M., Song, D.: Spatially transformed adversarial examples. In: Proceedings of the International Conference on Learning Representations (ICLR) (2018) 3
44. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2636–2645 (2020) 8
45. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017) 6