Quantifying the use of domain randomization

Ani, Mohammad; Basevi, Hector; Leonardis, Ales

Quantifying the Use of Domain Randomization

Mohammad Ani
School of Computer Science
University of Birmingham
Birmingham, UK
mxa563@cs.bham.ac.uk

Hector Basevi
School of Computer Science
University of Birmingham
Birmingham, UK
h.r.a.basevi@cs.bham.ac.uk

Aleš Leonardis
School of Computer Science
University of Birmingham
Birmingham, UK
a.leonardis@cs.bham.ac.uk

***Abstract*—Synthetic image generation provides the ability to efficiently produce large quantities of labeled data, which addresses both the data volume requirements of state-of-the-art vision systems and the expense of manually labeling data. However, systems trained on synthetic data typically under-perform systems trained on realistic data due to mismatch between the synthetic and realistic data distributions. Domain Randomization (DR) is a method of broadening a synthetic data distribution to encompass a realistic data distribution and provides better performance when the exact characteristics of the realistic data distribution are not known or cannot be simulated. However, there is no consensus in the literature on the best method of performing DR. We propose a novel method of ranking DR methods by directly measuring the difference between realistic and DR data distributions. This avoids the need to measure task-specific performance and the associated expense of training and evaluation. We compare different methods for measuring distribution differences, including the Wasserstein and Fréchet Inception distances. We also examine the effect of performing this evaluation directly on images and features generated by an image classification backbone. Finally, we show that the ranking generated by our method is reflected in actual task performance.**

I. Introduction

The use of synthetic data for solving computer vision tasks has gained popularity due to the ease and speed of generating large-scale synthetic datasets, compared to collecting and annotating real data. Synthetic data generation is especially beneficial in cases where collecting data in the real-world is challenging. For example, to gather data for an autonomous driving task, we would typically have to navigate a variety of environments under different weather conditions long enough to capture the range of cars on the road. This data acquisition process is extremely time-consuming and expensive.

Synthetic data generation allows us to produce substantial amounts of annotated examples, which are crucial when using data-hungry deep neural networks (DNNs) [1]. However, solely using synthetic images to train DNNs commonly leads to poor performance when deployed in the real-world. We generally attribute the poor performance to domain shift, where our training domain (synthetic) differs from our testing domain (real-world). Several works in domain adaptation have tackled this problem [2]–[5]. Recently, an inexpensive way to achieve transfer from synthetic-to-real, particularly in several robotics applications [6]–[14], is through Domain Randomization (DR).

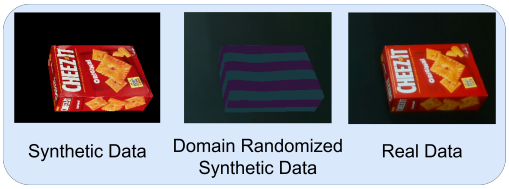

Fig. 1: Sample Data from Different domains

The key idea is to generate variation in the synthetic dataset by randomizing various simulator parameters such as textures, illumination, number of objects, object poses, or camera positions. The common intuition behind this is that during inference on real images, the real images would have similarities to some subset of the synthetic training distribution [7]. Fig. 1 shows the differences between synthetic, DR synthetic, and real-world data. Unlike some generative models, where the goal is to attempt to match the target distribution [15], DR is not trying to produce synthetic images indistinguishable from real-world images.

While this augmentation process of randomizing simulator parameters has demonstrated the potential for transfer from synthetic-to-real [7], [8], it is unclear which particular DR method is most appropriate for an arbitrary task.

Currently, there is no universal approach for DR. Take, for example, the task of predicting objects' positions on a table. Assuming we know the shape of the objects but not the textures, we can sample from a distribution of textures to augment our synthetic training data with varied textures to localize the objects.

While some would opt to randomize the objects' textures in a scene using flat RGB colors, others would use more complex patterns or additional noise applied to the textures. [11], [17], [19]–[21]. We expect the resulting network trained using DR synthetic data to perform well when evaluated using real-world data. However, the training process may be hindered by inefficiently sampling DR data.

We address this challenge by proposing a method for quantifying the differences between the DR synthetic and target distributions, where the synthetic samples are from distributions on the simulator inputs.

We achieved this by estimating the Wasserstein, and Fréchet Inception distances (FID) in the image and feature space, be-

Texture Randomization Techniques	[16]	[17]	[18]	[19]	[14]	[20]	[10]	[21]	[12]	[22]	[8]	[11]	[7]	[6]	[23]	[24]	[25]	[13]	[9]
Flat RGB	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Gradient RGB																			
Patterns (Checkerboard)							■	■				■		■					
Patterns (Striped)	■																		
Patterns (Other)											■						■		
Additional Noise (Perlin)			■						■			■							
Real Images	■				■													■	■

TABLE I: Table showing texture randomization techniques applied in current literature. The heavily favored approach is to use Flat RGB textures, in which each texture is a single RGB color sampled from a predetermined distribution.

tween realistic images and synthetic images that have modified textures based on the commonly used texture randomization methods in Table I.

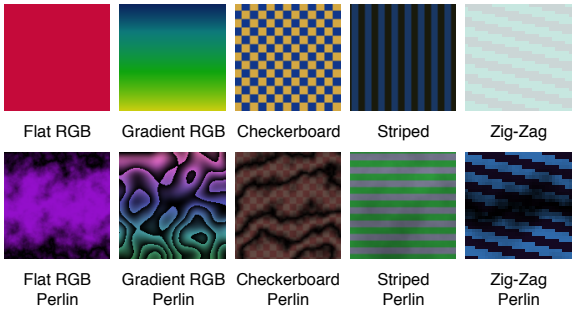

Fig. 2: Sample textures from our augmentation routine derive from the various techniques currently applied within DR literature from Table I.

The textures applied are non-patterned (Flat RGB, Gradient RGB), patterned (Checkerboard, Striped, Zig-Zag), and dominant noise (Perlin noise [26] applied to the previous textures), as seen in Fig. 2. By evaluating texture randomization techniques on a localization task, we show patterned textures result in the highest performance. We find that task performance, Wasserstein, and FID estimates in the feature space of ImageNet weights produce similar rankings.

Our main contributions are summarized below:

- We propose a novel method of quantifying differences between DR data distributions from samples, using neural networks.
- We demonstrate that the method is capable of ranking the different augmentations and is reflected in the performance of an object localization task.
- Based on the produced ranking, generated without any task-based training, we recommend using more complex patterned textures when generating DR synthetic data.

II. Related Work

Use of Synthetic Data. Since data-driven deep learning approaches started gaining popularity, generating synthetic data has been seen as a more efficient alternative to manually collecting and labeling real-world data. Synthetic datasets have been made available for solving various tasks such as autonomous driving [27]–[29], object detection [30], [31], segmentation [32]–[34], and robotic manipulation [9]. However, systems trained on existing synthetic datasets [35], [36] have shown limited ability to generalize to data that differs from the synthetic source, as the variety needed may not be present.

A practitioner may hand-design datasets for a task through the simulator and renderer configuration. However, realistic simulation can be computationally expensive [16], [36], [37]. Designing the environments typically requires expert knowledge of the domain to ensure that generated data is useful in solving the task. In this paper, we aim to utilize synthetic texture randomization routines and quantify the influence of the generated samples on a given task.

Domain Randomization and Domain Adaptation. Domain Adaptation has shown success at tackling the problem of adapting networks trained on a source domain to an unseen target domain [2], [38]–[41]. Several of these works make use of Generative adversarial networks' (GANs) [15] architecture to modify a given input's appearance.

For example, Shrivastava et al. [2] addressed synthetic data's unrealistic appearance by refining the synthetic images using unlabeled real-world images to enhance realism for solving gaze and hand pose estimation. Such an approach typically relies on having access to a large amount of real data, such as approximately 214K images from the MPIIGaze dataset [42] in the case of gaze estimation.

DR utilizes the rendering engine and simulator to randomize parameters such as textures, illumination, object positions, and camera positions, resulting in a greater variety in the dataset. This approach to augmenting the synthetic data is an inexpensive way to improve transfer from synthetic to real in tasks such as manipulation [7], [18], [20], [21], [23], [24], pose estimation [16], [43], object detection [11], [19], and segmentation [17].

The user typically selects the rendering parameters, which are commonly sampled from a uniform distribution. While this may eventually lead a given model to learn from a heavily augmented dataset, the naive sampling of the rendering parameters may result in an inefficient data distribution for a given task. This sampling approach would include very slight variations, and the addition of inefficient data would be computationally wasteful and may hinder performance.

Existing literature does not perform a thorough comparative analysis of potential augmentations and typically relies on ablation studies to demonstrate the effects on a given task. For example, Table I shows that most implementations apply Flat RGB, despite the range of possible augmentation techniques. We show that measuring the distance between distributions in an image classification feature space provides insights on

task-based performance – a novel approach in the context of DR.

Our work highlights the importance of selecting more favorable augmentations by establishing a correlation between different augmentations, measures of differences between data distributions, and localization task performance.

Measuring Distance Between Distributions. Quantifying distance between different distributions is an integral part of several machine learning processes, particularly in generative models. For example, the primary objective of Variational Autoencoders (VAEs) [44] and GANs [15], [45], [46] is to replicate a given data distribution.

VAEs use KL-Divergence in Equation (1) to measure the distance between two continuous probability distributions P and Q.

$$D_{KL}(P||Q) = \int_x P(x) log\left(\frac{P(x)}{Q(x)}\right) dx \tag{1}$$

$D_{KL} = 0$ when $P(x) = Q(x)$. As the KL-Divergence is asymmetric, an issue arises in the instance where $Q(x) \approx 0$ and $P(x) > 0$, as the distance measure may tend to infinity.

The standard GAN uses the Jensen-Shannon divergence between P_{data} (the original data distribution) and P_g (the model's generated distribution), both defined on a compact data space χ.

$$D_{JS}(P_{data}||P_g) = \frac{1}{2}D_{KL}\left(P_{data}||\frac{P_{data}+P_g}{2}\right) + \frac{1}{2}D_{KL}\left(P_g||\frac{P_{data}+P_g}{2}\right) \tag{2}$$

Jensen-Shannon Divergence (JSD) in Equation (2) being symmetric and bounded by $[0, 1]$ allows for smoother training for the generative models. However, if the distributions are far apart, the estimate is less meaningful as an indicator of the sample quality of a generator G [45]. The first implementation of the Wasserstein-GAN (WGAN) presented a comparison of established approaches to quantifying distance in distributions such as JSD and Wasserstein metric as a loss function. The comparison shows a correlation between lower error produced by the Wasserstein metric as a loss function and better sample quality from a given Generator G [45]. Because JSD saturates at $ln(2)$, it becomes less useful at measuring distances when the distributions become far apart.

The use of the Wasserstein distance in Equation (3) addresses JSD's issue as a less meaningful metric. Where $\Pi(P_{data}, P_g)$ denote the set of all joint distributions $\gamma(x, y)$ with marginals P_{data} and P_g, the Wasserstein distance is:

$$W(P_{data}, P_g) = \inf_{\gamma \in \Pi(P_{data}, P_g)} \mathbb{E}_{(x,y)\sim\gamma}\left[||x - y||\right] \tag{3}$$

To estimate the Wasserstein distance, the discriminator must be Lipschitz continuous. There are several different ways works have enforced a Lipschitz constraint on the discriminator, each with varying results. Initially, with WGANs, weight clipping enforced the constraint [45], while others implemented a gradient penalty [46]. Implementing the Lipschitz constraint on the discriminator remains an open problem, and in turn, affects the accuracy of the estimate. The variations in enforcing the Lipschitz constraint on the discriminator prevents direct quantitative comparison between individual methods. Here, we are interested in ranking the different texture randomization methods.

III. Method

This section discusses the approach taken for measuring the distances between distributions in the image space and distributions of extracted features. We describe how we validate our findings using an object localization task on realistic images.

To estimate the distance between distributions, we investigate the use of the Wasserstein and Fréchet Inception distances between two distributions. The following subsections describe the approaches taken for the image and feature space.

A. Image Space

With the Wasserstein distance, we use a modified implementation of WGAN-GP [46] and replace the generated samples with DR samples to the WGAN-GP critic. We use the computed Wasserstein distance to rank the various texture randomization methods from the lowest to highest distances.

We evaluate this ranking on an object localization task, where the goal is to predict the 3D position of an object of interest using VGG-16 [47]. In the image space, we use the same VGG-16 architecture implementation found in one of the seminal DR works by Tobin et al. [6].

B. Feature Space

Pre-trained backbones are widely available to bootstrap learning for new tasks. The availability of pre-trained backbones makes it possible to measure the distance between distributions in feature space from existing networks already trained on a large amount of data.

In the feature space, we estimate the Wasserstein and Fréchet Inception distances based on features extracted from the Conv5 block in a ResNet-50 [48] model. Fig. 3 shows the proposed approach for quantifying the distances between distributions. When exploring the feature space, we use ResNet-50 as the feature extractor as it is a widely accepted and robust model for extracting feature vectors [13], [17], [48], [49]. We show that the different texture randomization methods' ranking reflects the performance for solving an object localization task in the feature space.

Domain Randomization. In this work, we focus on texture randomization, as textures are considered one of the most important decision criteria in neural networks and the most heavily used in robotics DR applications. [6]–[8], [10]–[12], [14], [17]–[25], [50]. We implement all the methods used in current DR literature in Table I.

We use a custom simulator to perform physics simulation and rendering for our experiments, which allows us to generate physically plausible scenes, and control rendering parameters

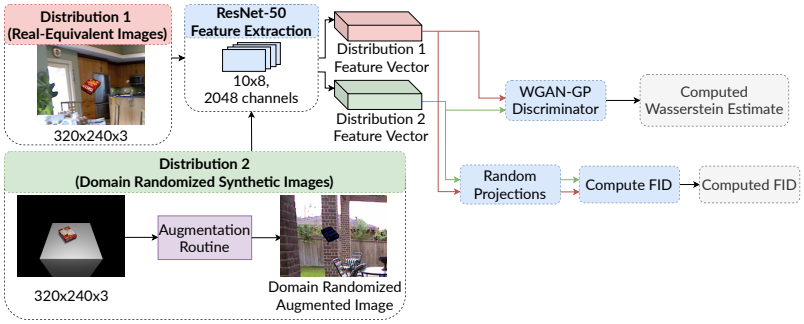

Fig. 3: Flow of data for estimating the Wasserstein distance and FID from two feature vector distributions.

during experimentation. We implemented the texture generation methods shown in Table I and can apply the resulting textures, such as those seen in Fig. 2, to the objects in scenes. Illumination is fixed, and object poses are shared across datasets, such that the differentiating factor between samples are the textures applied to the objects.

Quantifying Differences in Distributions. As we are dealing with unknown distributions when using raw images and in the feature space, we estimate the FID and Wasserstein distance using neural networks from distribution samples.

We use WGAN-GP instead of the original WGAN because of a more robust method of enforcing Lipschitz-Continuity [45], [46]. Due to the gradient penalty term in WGAN-GP restricting the norm of the gradient of the discriminator, we expect the implementation to affect the Wasserstein distance estimate, but not affect the ranking (ordering of the different texture randomization methods). We use a modified version to replace the generated samples with the randomized simulated data.

Quantifying Differences in Distributions – Feature Space. We subsequently modify the WGAN-GP's discriminator to take the input shape of each of the feature vectors tested.

As a comparison, we compute the distance between feature vectors between P_r and P_{aug} using FID [51]. FID is regularly used in generative models to measure the quality of the generated samples compared to the original distribution. Equation (4) shows the computation, where P_r is the real-textured, real-equivalent dataset, with mean and covariance (m_r, C_r) and P_{aug} is the augmented domain randomized synthetic dataset with mean and covariance (m_{aug}, C_{aug}).

$$d^2((m_{aug}, C_{aug}), (m_r, C_r)) = ||m_{aug} - m_r||_2^2 + \mathrm{Tr}(C_{aug} + C_r - 2(C_{aug}C_r)^{\frac{1}{2}}) \quad (4)$$

Using FID means we no longer need to train a discriminator to estimate the distance between distributions. However, we are assuming that the distributions are Gaussian and must process the entire dataset to estimate the covariance matrices, which can be computationally expensive for large feature vectors.

Localization Task. To validate the ranking generated by the Wasserstein distance and FID, we train an object detector to localize an object in a scene in terms of its 3D spatial position, (x, y, z). In the image space, we replicate the VGG-16 architecture from one of the original approaches to DR by Tobin et al. [6], [47] and in the feature space, we use a modified version of ResNet-50 [48]. The standard convolutional layers are used, with the addition of three fully connected layers. We use the MSE loss between the predicted object positions and ground truth using the Adam optimizer [52] with a learning rate of $1e-4$.

IV. Experiments

First, we compared real-equivalent synthetic and DR synthetic RGB images in the image space. Here, we considered the original, unmodified, textures to be our real-equivalent distribution. Next, we measured the distance between distributions in feature space in real-equivalent and DR synthetic images using real-world image backgrounds from the NYU Depth V2 dataset [53]. The NYU Depth V2 dataset comprises real-world RGB images from a variety of indoor scenes, and the addition of real-world image backgrounds forces the network to use the foreground rather than the background. The following subsections describe the experiments in further detail.

A. Quantifying Differences in Distributions in Image Space

For the synthetic datasets, we chose objects from the YCB dataset [54], as seen in Fig. 4. Using the YCB dataset gives us access to high-resolution scans of the objects' real-world texture and their meshes. This allowed us to modify desired parameters in the simulator.

We generated synthetic RGB images of size 320x240x3. The object of interest is the Cheez-It box, where the poses used are physically plausible and sampled from a Gaussian distribution of mean 0 and a standard deviation of 0.05 m around the center of the table. A sample of the synthetic (real-equivalent) and DR synthetic Cheez-It box dataset can be seen in Fig. 4. Each of the augmentation methods applied contains the same poses, camera position, background, and illumination. This ensures that the distance measured is related to the applied textures. We generated 3,000 images for the training set and 3,000 images for the test set for each of the augmentation methods applied. Colors and original patterns

between Perlin and non-Perlin textures are shared. For example, the same initial colors for Flat RGB are used for Flat RGB Perlin. This is to analyze the effects of dominant noise on the initial textures used.

To quantify the difference in distributions in the image space, we used WGAN-GP to estimate the Wasserstein distance on raw RGB images, where the inputs to the WGAN-GP critic are the real-equivalent synthetic data (assumed to be our real distribution here) and the DR synthetic data.

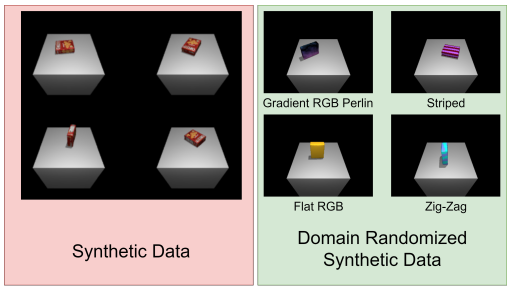

Fig. 4: Sample of the synthetic data (real-equivalent) and DR synthetic data used. Poses are shared across datasets, with the main differentiating factor being the textures applied to the Cheez-It box. The camera position, illumination, and background remain fixed.

Results from Quantifying Differences in Distributions in Image Space. Results from estimating the Wasserstein distance from the ten different randomization techniques on raw RGB images are shown in Fig. 5.

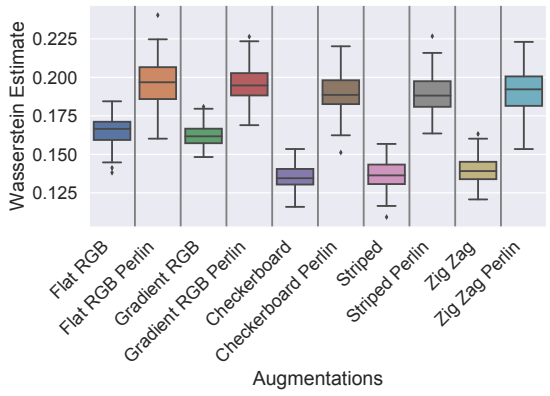

Fig. 5: Wasserstein distance estimate using different texture randomization techniques. We compute the estimate between real-equivalent synthetic and DR synthetic RGB images with black backgrounds. We see three distinct groupings between patterned (Checkerboard, Striped, Zig-Zag), non-patterned (Flat RGB and Gradient RGB) and dominant noise (Perlin).

Here, we find three distinct groupings between non-complex (Flat RGB and Gradient RGB), complex (Checkerboard, Zig-Zag, Striped), and dominant noise (Perlin noise). This suggests that the type of texture augmentation affects the distance estimates on raw images. To determine whether these differences are meaningful, we train on a localization task to see if the same rankings hold.

We trained a localization network on real-equivalent synthetic images and DR synthetic images with a modified VGG-16 architecture to use 3 FC layers. The network was trained with a batch size of 50, a learning rate of $1e-4$, and Adam optimizer [52]. The results from the task-based network are shown in Fig. 6.

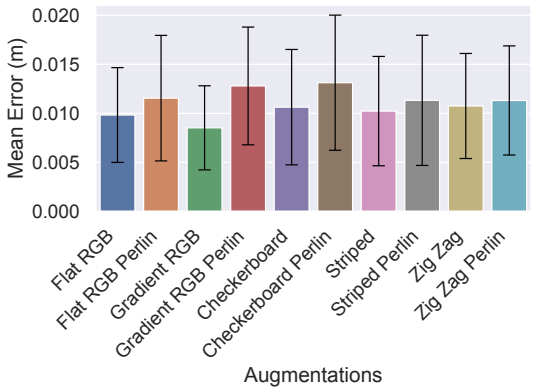

Fig. 6: Localization task when trained on DR synthetic images, and tested on real-equivalent synthetic images with black backgrounds. MSE is between the predicted and ground truth positions of the object on the table.

In this instance, the selection of augmentations for fixed, static backgrounds, illumination, and camera had little impact. The high variance means it is not possible to distinguish between rankings as clearly as in the Wasserstein distance estimates. We see that the localization task's mean error follows the same ranking as the Wasserstein estimate, with similar groupings based on the means only. However, the variance is too large to conclude a strong correlation between the Wasserstein estimate and the mean error of the localization task. Given the datasets used, and the task at hand, the augmentation techniques applied would perform similarly.

Realistic Backgrounds. We explored quantifying the differences in distributions using real-world backgrounds from the NYU Depth V2 dataset [53]. The modification replaced previous black backgrounds with real-world indoor scenes from the NYU Depth V2 dataset. Previous synthetic datasets were re-used to keep consistent poses and texture randomizations across previous experiments. No backgrounds are shared between the real-equivalent and DR synthetic datasets. A sample of the modified datasets can be seen in Fig. 7.

When estimating the Wasserstein distance using the modified real-world backgrounds on raw images, we do not get clear separations in ranking based on the augmentation techniques used, as seen in Fig. 8. Furthermore, there is a more significant variance in the distance estimates compared to the fixed black background datasets. We find that estimating

(a) Real-equivalent Data (b) DR Synthetic - Flat RGB

Fig. 7: Sample images from the real-image background dataset. The previous black backgrounds have been replaced with backgrounds from NYU Depth V2 dataset to lessen learning information based on the background rather than the foreground.

the Wasserstein distance on raw RGB images, with real-world backgrounds, to be less reliable in providing a clear ranking between the different texture randomization techniques.

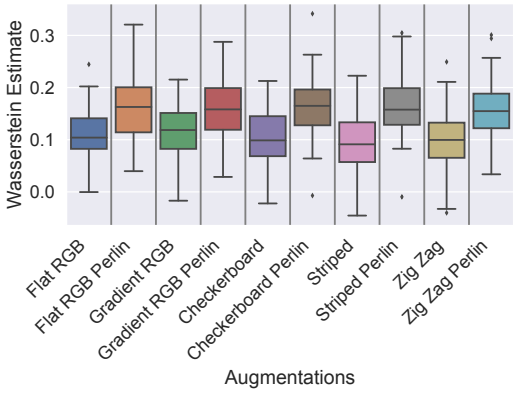

Fig. 8: Wasserstein estimate does not provide clear separations in augmentation techniques when operating with raw images using real-world backgrounds. We are only able to clearly distinguish between techniques involving non-Perlin and Perlin noise.

B. Quantifying Differences in Distributions in Feature Space

Operating on raw RGB images and real-world backgrounds did not appear to provide a clear ranking when applying the different texture randomization techniques. We explored quantifying the differences in distributions between two feature vectors from the same modified real-world background datasets, via Wasserstein distance and FID.

Extracting Features We investigated the effects of measuring the distance between distributions in feature space by using a ResNet-50 model [48], and analyzed the Wasserstein estimates from the Conv5 layer, before flattening, of the network.

Given the dataset and representation size, computation on FID is expensive. To reduce computational expense, we reduced the dimensionality for 3000 samples using 10 random projections to get the variance of the estimates; this may result in some loss of information. Each of the augmentation techniques uses the same 10 projection matrices to ensure fair comparisons.

Results from Quantifying Differences in Distributions in Feature Space. When looking at the feature space at the Conv5 block, using pre-trained ImageNet weights, we see more apparent separations between the various texture randomization techniques applied, as shown in Fig. 9. The ranking of augmentation techniques is more easily determined.

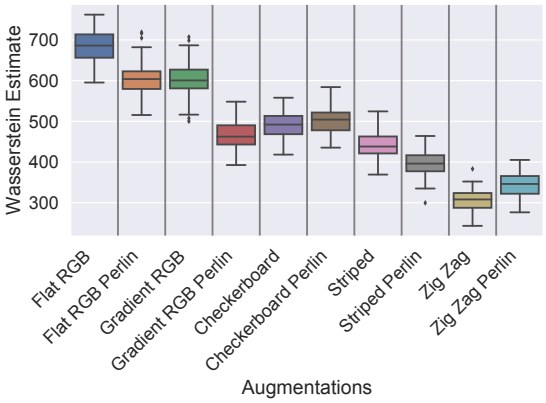

Fig. 9: Wasserstein distance between real-equivalent and DR synthetic data with backgrounds from NYU V2 dataset when operating in the feature space. The distance is measured using distributions of feature vectors extracted from a ResNet-50 backbone. When operating in the feature space, we are able to more clearly distinguish between the various augmentation techniques.

Comparison with Localization Task. When training the localization task for each of the augmentation techniques using random weights, we see more distinct separations between them. Here, we used the ResNet-50 network for the task, with the addition of 3 FC layers. We performed a z-test to evaluate statistical significance, with a null hypothesis that the mean is equal between two given augmentation techniques at $\alpha = 0.05$. The test yields a p-value of < 0.05 for Zig-Zag and Striped with Perlin, and a p-value < 0.001 for the remaining augmentation pairs.

After ranking the localization task performance and comparing it to the FID and Wasserstein distance, we see a correlation between task performance and estimates in the feature space at the Conv5 block, as shown in Fig. 10. This is particularly the case when comparing task performance with FID, showing less variance in the estimate. Both Wasserstein and FID produce clear distinctions between complex and non-complex textures, and selecting a more complex texture with lower estimates, results in better task performance.

With the addition of Perlin noise, the estimates generally follow the same trend, though the introduction of noise increases the difficulty in quantifying the underlying distributions, as shown in Fig. 10. In the cases of Zig-Zag Perlin and Gradient RGB Perlin, for example, both Wasserstein and FID rankings disagree with the localization ranking. This disagreement indicates that the approach is less robust when comparing task performance with textures containing Perlin noise.

Our findings are consistent with those of Borrego et al. [11], where the authors found that the use of more complex textures such as patterns in the texture augmentations used for object detection aids performance, while the use of non-complex textures such as Flat RGB provides the least impact on improving performance.

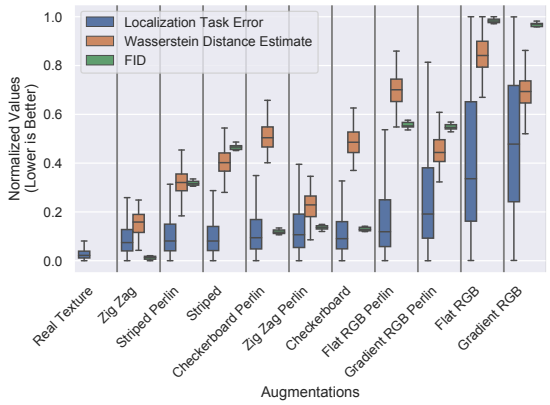

Fig. 10: Comparison of localization task, Wasserstein and FID estimate in feature space. Values are normalized and sorted by lowest mean error in the localization task. Effects of additional noise increases the difficulty in obtaining a clear ranking. In general, the addition of dominant Perlin noise appears to aid performance, as well as using patterned textures.

V. Discussion

Given a collection of commonly applied texture randomization methods, we quantify the differences in the distributions of the feature space and find clear rankings between the techniques when working with augmentations without additional noise as shown in Fig. 10. We rank methods by Wasserstein distance/FID estimates and by task error. We find that low Wasserstein and FID estimates are associated with lower task error. By Wasserstein/ FID, Zig-Zag (patterned) is best (lowest ranking) and Flat RGB (non-patterned) is worst. Sorting by mean task error, Zig-Zag is best (lowest task error), and Gradient RGB is worst.

We show that selecting the augmentation according to the Wasserstein ranking produces the best task performance for object localization. In this case, we recommend that practitioners use Zig-Zag textures for augmentation. Patterned textures are more favorable as opposed to non-patterned when randomizing the objects' textures, which is reflected in the final task performance.

However, we find that the addition of dominant Perlin noise increases the difficulty in attaining a clear ranking, particularly with FID. We see the FID estimates amongst patterned and non-patterned are similar, which may allude to the dominant noise overpowering the underlying texture pattern.

Here we demonstrated an effective means of ranking DR methods by quantifying differences between data distributions from samples using an estimate of the Wasserstein distance and FID. Furthermore, we predicted the final task ranking with good agreement, without task-based training and evaluation. This technique is not specific to a particular task, and could be used as a criterion for selecting randomization types for novel tasks given a small amount of real data.

VI. Acknowledgement

We acknowledge MoD/Dstl and EPSRC for providing the grant to support the UK academics involvement in a Department of Defense funded MURI project through EPSRC grant EP/N019415/1.

References

[1] S. I. Nikolenko, "Synthetic data for deep learning," 2019, arXiv:1909.11512.

[2] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-January, 2017, pp. 2242–2251.

[3] Y. Taigman, A. Polyak, and L. Wolf, "Unsupervised cross-domain image generation," in *5th International Conference on Learning Representations, ICLR*, 2017.

[4] W. Hong, Z. Wang, M. Yang, and J. Yuan, "Conditional Generative Adversarial Network for Structured Domain Adaptation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR*, 2018, pp. 1335–1344.

[5] P. O. Pinheiro, "Unsupervised Domain Adaptation with Similarity Learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR*, 2018, pp. 8004–8013.

[6] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, 2017, pp. 23–30.

[7] J. Tobin, L. Biewald, R. Duan, M. Andrychowicz, A. Handa, V. Kumar, B. McGrew, A. Ray, J. Schneider, P. Welinder, W. Zaremba, and P. Abbeel, "Domain randomization and generative models for robotic grasping," *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, pp. 3482–3489, 2018.

[8] F. Sadeghi and S. Levine, "CAD2RL: Real Single-Image Flight without a Single Real Image," in *Robotics: Science and Systems Conference (R:SS)*, 2017.

[9] M. Yan, S. Tyree, and J. Kautz, "Sim-to-Real Transfer of Accurate Grasping with Eye-In-Hand Observations and Continuous Control," in *Neural Information Processing Systems (NeurIPS) Workshop on Acting and Interacting in the Real World: Challenges in Robot Learning*, 2017.

[10] 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," in *CVPR Workshop on Autonomous Driving*, 2018.

[11] J. Borrego, A. Dehban, R. Figueiredo, P. Moreno, A. Bernardino, and J. Santos-Victor, "Applying domain randomization to synthetic data for object category detection," 2018, arXiv:1807.09834.

[12] S. James, A. J. Davison, and E. Johns, "Transferring End-to-End Visuomotor Control from Simulation to Real World for a Multi-Stage Task," *1st Conference on Robot Learning (CoRL)*, 2017.

[13] A. Prakash, S. Boochoon, M. Brophy, D. Acuna, E. Cameracci, G. State, O. Shapira, and S. T. Birchfield, "Structured domain randomization: Bridging the reality gap by context-aware synthetic data," *International Conference on Robotics and Automation (ICRA)*, pp. 7249–7255, 2019.

[14] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke, "Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping," in *International Conference on Robotics and Automation (ICRA)*, 2018.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems (NeurIPS) - Volume 2*, 2014, p. 2672–2680.

[16] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects," in *Conference on Robot Learning (CoRL)*, 2018.

[17] D. Ward, P. Moghadam, and N. Hudson, "Deep Leaf Segmentation Using Synthetic Data," in *British Machine Vision Conference (BMVC) Workshop on Computer Vision Problems in Plant Pheonotyping (CVPPP)*, 2018.

[18] J. Matas, S. James, and A. J. Davison, "Sim-to-Real Reinforcement Learning for Deformable Object Manipulation," in *Conference on Robot Learning (CoRL)*, 2018.

[19] M. Sundermeyer, "Implicit 3D Orientation Learning for 6D Object Detection from RGB Images," in *European Conference on Computer Vision (ECCV)*, 2018.

[20] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense Object Nets: Learning Dense Visual Object Descriptors By and For Robotic Manipulation," in *Conference on Robot Learning (CoRL)*, 2018.

[21] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric Actor Critic for Image-Based Robot Learning," in *Robotics: Science and Systems (R:SS)*, 2018.

[22] J. Tremblay, T. To, A. Molchanov, S. Tyree, J. Kautz, and S. T. Birchfield, "Synthetically trained neural networks for learning human-readable plans from real-world demonstrations," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–5, 2018.

[23] OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research (IJRR)*, vol. 39, no. 1, pp. 3–20, 2020.

[24] F. Zhang, J. Leitner, Z. Ge, M. Milford, and P. Corke, "Adversarial discriminative sim-to-real transfer of visuo-motor policies," *The International Journal of Robotics Research (IJRR)*, vol. 38, no. 10-11, pp. 1229–1245, 2019.

[25] S. Pouyanfar, M. Saleem, N. George, and S.-C. Chen, "Roads: Randomization for obstacle avoidance and driving in simulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[26] K. Perlin, "Improving noise," in *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '02. ACM, 2002, pp. 681–682.

[27] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3234–3243.

[28] 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)*, vol. 9906, 2016, pp. 102–118.

[29] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning (CoRL)*, 2017, pp. 1–16.

[30] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," *IEEE International Conference on Computer Vision (ICCV)*, pp. 1310–1319, 2017.

[31] G. Georgakis, A. Mousavian, A. C. Berg, and J. Kosecka, "Synthesizing training data for object detection in indoor scenes," *ArXiv*, vol. abs/1702.07836, 2017.

[32] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid, "Learning from synthetic humans," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4627–4635, 2017.

[33] Y. Chen, W. Li, X. Chen, and L. Van Gool, "Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1841–1850.

[34] S. Sankaranarayanan, Y. Balaji, A. Jain, S. N. Lim, and R. Chellappa, "Learning from Synthetic Data: Addressing Domain Shift for Semantic Segmentation," *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[35] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4340–4349, 2016.

[36] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation?" *IEEE International Conference on Computer Vision (ICCV)*, pp. 2697–2706, 2017.

[37] C. Jiang, S. Qi, Y. Zhu, S. Huang, J. Lin, L.-F. Yu, D. Terzopoulos, and S.-C. Zhu, "Configurable 3d scene synthesis and 2d image rendering with per-pixel ground truth using stochastic grammars," *International Journal of Computer Vision (IJCV)*, vol. 126, no. 9, p. 920–941, Jun 2018.

[38] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 700–708.

[39] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *International Conference on Machine Learning (ICML)*, 2018.

[40] Y. Zou, Z. Yu, B. V. K. V. Kumar, and J. Wang, "Domain adaptation for semantic segmentation via class-balanced self-training," in *European Conference on Computer Vision (ECCV)*, 2018.

[41] S.-W. Huang, C.-T. Lin, S.-P. Chen, Y.-Y. Wu, P.-H. Hsu, and S.-H. Lai, "Auggan: Cross domain adaptation with gan-based data augmentation," in *European Conference on Computer Vision (ECCV)*, 2018.

[42] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, "Appearance-based gaze estimation in the wild," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4511–4520, 2015.

[43] X. Ren, J. Luo, E. Solowjow, J. A. Ojea, A. Gupta, A. Tamar, and P. Abbeel, "Domain randomization for active pose estimation," *International Conference on Robotics and Automation (ICRA)*, pp. 7228–7234, 2019.

[44] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, (ICLR)*, 2014.

[45] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, 2017, pp. 214–223.

[46] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, p. 5769–5779.

[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR,*, 2015.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[49] S. Hinterstoisser, V. Lepetit, P. Wohlhart, and K. Konolige, "On Pre-Trained Image Features and Synthetic Images for Deep Learning," 2017, arXiv:1710.10710v2.

[50] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness." in *International Conference on Learning Representations (ICLR)*, 2019.

[51] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *International Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[52] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, vol. abs/1412.6980, 2015.

[53] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision (ECCV)*, 2012.

[54] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Yale-CMU-Berkeley dataset for robotic manipulation research," *International Journal of Robotics Research (IJRR)*, vol. 36, no. 3, pp. 261–268, 2017.