Evaluating Text-to-Image Diffusion Models for Texturing Synthetic Data

Thomas Lips Francis wyffels

AI and Robotics Lab, IDLab, Ghent University - imec

{thomas.lips,francis.wyffels}@ugent.be

Abstract

Building computer vision systems that can handle diversity in objects or environments often requires large amounts of data, which can be difficult to collect. Synthetic data generation offers a promising alternative, but limiting the sim-to-real gap requires significant engineering efforts. To reduce this engineering effort, we investigate the use of pretrained text-to-image diffusion models for texturing synthetic images. In particular, we compare diffusion-based texturing with using random textures, a common domain randomization technique in synthetic data generation. We evaluate the texturing approaches on two object-centric representations: keypoints and segmentation masks and measure their efficacy on real-world datasets for three object categories: shoes, T-shirts, and mugs. Surprisingly, we find that texturing using a diffusion model performs on par with random textures, despite generating seemingly more realistic images. Our results suggest that, for now, using diffusion models for texturing does not provide advantages over the conceptually simpler method of using random textures.

1. Introduction

To deal with the typical diversity in environments and object shapes or appearances, learning-based methods are often required when building computer vision systems. It is well established that the performance of these learned models strongly depends on the amount of data available to train them. However, real-world data collection requires large efforts [1, 15, 17]. Therefore, the amount of data is often a bottleneck in the creation of generic computer vision systems. Pretrained foundation models reduce the need for task-specific data, but have not yet completely overcome this need [37].

A parallel approach to overcome this data bottleneck is to train the system on synthetically generated data instead of real-world data. The main difficulty with synthetic data is to ensure that models trained on synthetic data transfer well to the real world, i.e., to limit the sim-to-real performance gap [23]. In practice, this often requires significant amounts of manual engineering for 3D asset generation, scene composition and texturing [14, 20, 26, 34]. In this work, we focus on texturing, which can be summarized as creating the appearance of 3D objects and scenes by specifying the optical properties (color being the most important one) of each part of an object. Recently, researchers have sought to (partially) outsource synthetic data generation to neural networks, for example, by generating synthetic images using text-to-image diffusion models [4, 21, 36].

2. Related Work

We further investigate the use of text-to-image diffusion models to texture RGB images of a 3D scene and compare this method against using random textures. To generate the synthetic images, we first create a 3D scene and obtain the annotations for that scene. We then texture the scene by either adding random textures to all elements or by using a diffusion model to generate the textures. The process is illustrated in Figure 1.

We evaluate the texturing approaches on two pixel-level and object-centric representations: keypoint detection and segmentation. These are often used in computer vision systems that require high precision, including robotic manipulation, which we use as an application context [24, 40]. These representations require precise annotations, and therefore we first create an explicit 3D scene instead of directly generating images from text prompts using a text-to-image diffusion model: From the 3D scene, we can extract pixel-perfect annotations. For this approach to work, we need to ensure that the diffusion model does not alter the semantics of the scene during texturing, as this would invalidate the annotations. For example, the diffusion model cannot alter the shape of the object or change its pose in the image. To accomplish this, we use a Controlnet [38] to condition on both a depth image of the scene and a prompt, as in [4] and [22].

We evaluated the efficacy of the data generation methods by measuring the downstream performance of models trained on the data. We generated data for static scenes of 3 different object categories: shoes, T-shirts and mugs. The models were evaluated on real-world test datasets using

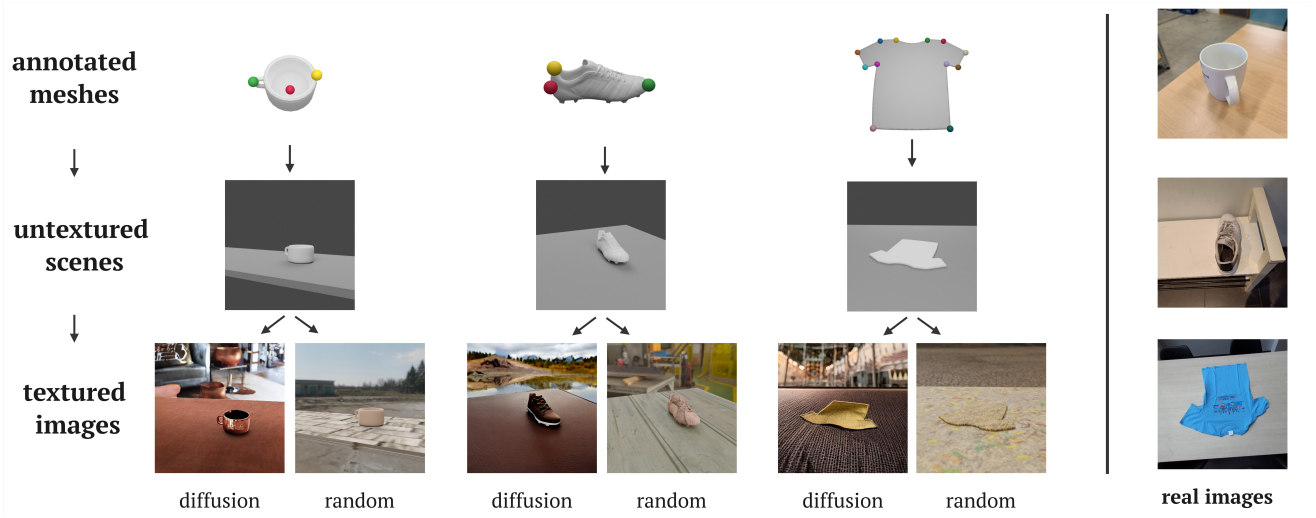

Figure 1. Left: In this work, we compare text-to-image diffusion models against random textures for texturing 3D scenes in a synthetic data generation pipeline. Right: We evaluate the efficacy of the synthetic data on real-world data for both keypoint detection and segmentation.

common metrics for each representation: mean average precision (mAP) [19] for segmentation and average keypoint distance (AKD) for keypoint detection [32].

Surprisingly, we found that texturing with a diffusion model performs similarly to using random textures. In a series of additional experiments, we observed that both methods exhibit limited scaling behavior and that using LLM-generated prompts resulted in the best performance for diffusion-based texturing.

In summary, our contributions are as follows.

- We developed a data generation pipeline to learn fine-grained object representations leveraging text-to-image diffusion models
- We extensively compare diffusion-based texturing with using random textures, and surprisingly find that they perform similarly.
- We provide insight into the use of diffusion models for synthetic data generation by analyzing the scaling behavior and comparing different prompt generation methods.

2.1. Synthetic Data Generation

Synthetic data generation offers a compelling alternative to manual data collection for supervised machine learning. It provides arbitrary amounts of perfectly labeled data, enabling the desired generalizations. The main challenge is to overcome the sim-to-real gap to ensure that models trained on synthetic data generalize to real-world scenarios [23]. Common strategies to overcome this gap include domain randomization [31], in which the appearance and shape of objects, or the composition of the scene is varied beyond what is considered realistic, and domain adaptation [12], in which the differences between synthetic and real data are explicitly learned. Despite these recipes, achieving strong sim-to-real performance often requires significant human effort to improve the diversity and quality of assets (object shapes, materials) and scene compositions used for data generation [14, 26, 34].

Researchers have tried to reduce human effort using generative models. For example, [2] uses a class-conditioned GAN [10] to train classifiers on images generated by the GAN. [39] go beyond image-level semantics and trains a decoder on the latent codes of a GAN to automatically obtain segmentation masks and keypoints for generated images. However, the quality of the images generated by such GANs is limited. Furthermore, these GANs must be explicitly trained on each category. Recently, large, pretrained text-to-image diffusion models [13, 28] have been explored to overcome these limitations.

2.2. Text-to-image diffusion models for synthetic data

Text-to-image diffusion models [28] have been used to generate synthetic data for image classification [7, 9, 22, 30], semantic segmentation [21, 25, 35], 3D pose estimation [22] and robot trajectory augmentation [4, 36].

For image classification, [7] show that diffusion-generated synthetic data does not scale as well as real data. [9] showed that directly training on the underlying dataset of the generative model can outperform training on synthetic images generated by the diffusion model.

For segmentation, a pixel-perfect object mask is required, in addition to controlling the object category with a textual description. [35], [21] and [25] use the cross-attention between text and images in Stable Diffusion [27]

to generate these masks automatically. [25] uses self-attention to improve the generated masks and generates multiclass annotations.

For pose estimation, [22] uses 3D meshes to generate edge maps and then renders images for these edge maps using a Controlnet. They also report results for segmentation and classification, improving on previous methods that do not use explicit 3D control.

For data augmentation, [36] augments robot trajectories by inpainting parts of the image. [4] first renders depth images of an object and then uses a Controlnet [38] to texture them, after which they are used to augment the robot trajectories.

In [4] and [22], a Controlnet [38] is used to condition on both text prompts and renders of 3D objects to increase control over the semantics of the generated images.

Various prompt strategies have been explored for diffusion models, including fixed templates [7], generated image captions [7, 35] and LLM-generated prompts [21, 22].

In this work, we require precise, pixel-level annotations. We, therefore, follow [4] and [22] and condition on both text prompts and renders of 3D scenes using a Controlnet with Stable Diffusion. Our work is closely related to [21], but we consider keypoint detection, make a more extensive comparison with using random textures, and compare various prompt generation methods.

3. Synthetic Data Generation

In this work, we generate synthetic images of static scenes that contain an object on a table. Learning representations on such images enables robotic manipulation of the object category, and this is the motivation for our work.

The data generation process consists of two steps: In the first step, we gather 3D meshes, annotate these meshes and use them to generate 3D scenes. In the second step, we texture the scene to provide the desired visual diversity. Combining the annotations from the first step with the textured images obtained after the second step, we obtain a diverse dataset for the object category with pixel-accurate labels.

For the second step, we compare different approaches to texture the scene, using either random textures, a common technique in domain randomization [29], or using text-to-image diffusion models. Both stages are described in more detail in the following sections. Figure 1 illustrates the data generation process used in this work.

3.1. Scene Generation

For each category of objects for which we want to create synthetic data, we first need to acquire a set of meshes. The meshes do not need to be of a very high quality and in particular do not require accurate UV-maps, which are often hard to get. In addition to gathering the meshes, we also need to obtain the required annotations. In this work, these are the 3D positions of the semantic keypoints and the object masks. The object masks can be simply obtained from the rendering engine. The keypoints can be manually annotated for each mesh, but it is often possible to determine them automatically based on the geometry of the mesh.

Once we have the meshes and their annotations, we generate 3D scenes of the objects. To model the table, we simply use a 2D plane. To introduce the desired scene geometry variations, we randomize the table's dimensions as well as the object and camera pose.

In these scenes, as we know the intrinsics and extrinsics of the camera, we can project all 3D mesh annotations to the image planes, obtaining pixel-perfect annotations. To generate visual diversity, we also need to texture the scene, which is discussed next.

3.2. Texturing

To texture (an image of) the scene, we consider two different approaches: In the first approach, we simply apply random textures to the elements of the scene. In the second, we use a text-to-image diffusion model and condition it on a depth image of the 3D scene and a suitable prompt. Each approach is now discussed in more detail.

3.2.1. random textures

With this method, we apply a random texture to the meshes of the object and the surface. In addition, we use a 360 image as a scene background to further increase visual diversity. We follow [20] and use textures and 360 images from PolyHaven[1].

3.2.2. diffusion texturing

We use a depth-conditioned text-to-image diffusion model to texture the scene. In fact, to be more precise, we texture a 2D image of the 3D scene. To do so, we first generate a list of descriptions for both the object's appearance and plausible scene backgrounds. These descriptions then serve as input to the diffusion model, together with a depth image of the scene, taken from the desired camera pose. The diffusion model then outputs an RGB image of the scene.

By also conditioning on a depth image, we make sure that texturing does not alter the semantics of the object, ensuring the accuracy of the precomputed 2D annotations. We use Controlnet [38] for this image conditioning and use the Stable Diffusion 1.5 [28] text-to-image model throughout this work.

4. Experiments

We evaluated the data generation procedures described in Section 3 on three object categories: mugs, shoes and T-shirts. We generated a dataset for each category and trained models for two object representations: keypoint detection

[1] https://polyhaven.com/

and segmentation masks. For all experiments, we report the performance of these models on our real-world test datasets. Section 4.1 provides more details about the synthetic and real datasets. The tasks and the metrics used to evaluate them are introduced in Section 4.2. The remaining sections describe the experiments we conducted, comparing diffusion-based texturing with the use of random textures in Section 4.3 and further exploring different aspects of the diffusion-based texturing pipeline in Section 4.4.

4.1. Object categories & datasets

We evaluated three object categories: mugs, shoes, and T-shirts. For each category, we generated synthetic data using the methods described in Section 3. For the mugs, we gathered 100 meshes from the Objaverse [5] dataset. 214 shoe meshes were obtained from the Google Scanned Objects dataset [6]. For the T-shirts, we used 250 meshes from [20]. For each category, 2500 distinct 3D scenes were generated by varying the mesh pose, the size and orientation of the table and the camera pose. Fig 1 shows a number of meshes and generated 3D scenes. 5000 images were generated from these scenes by sampling different camera poses for each scene and texturing them using one of the methods described in Section 3.2. We used Blender [3] to generate scenes and random texture datasets. To create the diffusion textures, we used Huggingface Diffusers [33]. All hyperparameters for the diffusion models were set to their default values, except for the *conditioning scale*, which we set to 1.5 to ensure that the semantics remained unchanged during texturing. Using an NVIDIA RTX3090 GPU, it took about 3s to render a 512x512 image with random textures using Cycles, Blender's physically-based renderer. Running inference on the diffusion model for texturing also took around 3s per image.

We evaluated the performance on a real-world test dataset and also provided a baseline train dataset with real images to put the results in perspective. For the T-shirts we used the aRTF dataset from [20], for the mugs and shoes we collected and annotated datasets manually: For the evaluation dataset we gathered a set of mugs and shoes and took pictures with a smartphone in various backgrounds. We gathered another set of mugs and shoes for the training dataset, but this time used a robot to auto-collect images from various angles. Backgrounds and objects are distinct in the train and test splits, to properly measure generalization. All images were manually annotated. The dataset sizes and number of distinct objects are given in Table 1. The number of objects is similar to [24]. The number of training images is about an order of magnitude smaller and more training images would likely increase the performance of the real-world baseline. Fig. 1 shows images from the real datasets on the right.

Table 1. Number of images and unique objects used in the real-world evaluation and baseline datasets.

	train dataset		**evaluation dataset**	
category	# images	# objects	# images	# objects
Mugs	1500	21	350	15
Shoes	2000	15	300	15
Tshirts	210	15	400	20

4.2. Performance Evaluation

We used two different tasks to evaluate the performance of the synthetic data: semantic keypoint detection and instance segmentation. Both require precise annotations and are often used in robotics [20, 24, 32]. For each task, we briefly discuss the training setup and the metric used to evaluate performance in the following sections. We refer to the accompanying code repository for more details.

In addition to measuring the task performance of models trained on the synthetic data, which is expensive, we have tried common image metrics such as CLIP-score [11] to quantify the quality of the dataset, but found that these correlate very poorly with the downstream task performance and therefore do not report them in this paper.

4.2.1. Keypoint Detection

Following [20], we formulate keypoint detection as pixel-wise regression of 2D target heatmaps. Each semantic category is mapped onto a different heatmap. Ground truth heatmaps are generated from the annotations by creating a Gaussian blob around each ground truth keypoint. The predicted heatmaps are regressed to the ground truth heatmaps using a binary cross-entropy loss.

To measure performance, we used the average keypoint distances (AKD), also known as RMS, between the ground truth keypoints and the predicted keypoint with the highest probability [32].

For the T-shirts, we used the same 12 keypoints as in [20]. For the shoes, we defined 3 keypoints on the nose, heel and tip. For the mugs, we defined 3 keypoints on the handle, bottom and top rim. These keypoints differ slightly from [24], as we found it easier to annotate keypoints that are on the surface of the object. The keypoints are visualized in Figure 1.

4.2.2. Instance Segmentation

For instance segmentation, we used YOLOv8 [16]. All hyperparameters are set to their default value, and we use the small model variant, pretrained on the COCO dataset [19]. To measure performance, we report the mean average precision (mAP) on different IoU thresholds ranging from 0.5 to 0.95, which is the default segmentation metric for COCO [19].

Table 2. Performance of the different texturing methods for all object categories and tasks. Random textures perform similar to diffusion textures. Both outperform the real baseline.

	keypoint AKD(↓)			**segmentation mAP(↑)**		
Training dataset	Mugs	Shoes	Tshirts	Mugs	Shoes	Tshirts
real data baseline	21.7	33.7	**25.6**	0.97	0.88	0.87
random textures	18.3	**13.4**	37.9	0.97	0.94	0.75
diffusion texturing	**17.4**	19.6	45.8	**0.99**	**0.95**	**0.93**

4.3. Comparing Texturing Methods

We now compare the performance of synthetic data generated with random textures against data textured using a diffusion model, as described in Section 3.2. The performance of the synthetic data generated by the different texturing methods is given in Table 2. We also provide the performance of a real-world train dataset as a baseline. We observed that both random textures and diffusion textures outperform the real data baseline in most cases, confirming the efficacy of our synthetic data pipelines. Comparing both texturing approaches, we observed that diffusion textures perform very similar to random textures, which is surprising as the images obtained through diffusion texturing seem more realistic (see Fig. 1 for examples).

4.4. Further Exploration of Diffusion Texturing

Next to comparing diffusion texturing against random textures, we have performed additional experiments to validate some design choices and to provide additional insight. We compared different strategies to generate prompts for the diffusion models and evaluated the scaling behavior of both methods. These experiments and their results are described in this section.

4.4.1. Prompting Strategy

A key design choice when using text-to-image diffusion models is how to prompt the models. In this experiment, we compared three different prompting strategies.

The first and simplest strategy is to use a fixed caption for each category, e.g., *A photo of a shoe*.

To create diverse prompts and thereby more diverse images, we also used a BLIP [18] model to caption images from the real training sets for each category. We then used these captions as prompts for the diffusion model. This method aims to match the prompts with the real (target) images. We collected approximately 3000 prompts for each category using this strategy.

For the final strategy, we queried an LLM (we used Google Gemini) to generate descriptions using the following prompt: *provide a description for X. Include color, patterns, materials and other visual characteristics.*. We randomly combined descriptions for the object and the table, obtaining a set of 5000 prompts for each category.

Table 3. Comparison of different prompting strategies for diffusion texturing. Using prompts generated by an LLM produced the best results.

	AKD(↓)			**mAP$_{seg}$(↑)**		
strategy	Mugs	Shoes	Tshirts	Mugs	Shoes	Tshirts
classname	22.8	23.4	66.4	0.98	0.90	0.77
BLIP captions	**16.3**	25.2	45.9	**0.99**	0.94	0.90
LLM prompts	17.4	**19.6**	**45.8**	**0.99**	**0.95**	**0.93**

For each prompting strategy, we generated 5000 images using the diffusion texturing pipeline and trained models on these datasets for both tasks. The results are provided in Table 3. Using a fixed template performed worse than using BLIP captions or LLM-generated prompts. The LLM-prompts scored slightly better than the BLIP captions. In addition, using LLM-prompts does not require real target images making this strategy more flexible. Therefore, we used the LLM-generated prompts in all other experiments of the paper.

Our findings are in line with [7], where the authors also found fixed templates inferior to BLIP captions. LLM-based prompts are a.o. used in [21] and [22], but to the best of our knowledge, they have not been explicitly compared with other prompting strategies for synthetic data generation.

4.4.2. Data Scaling Behavior

We have also explored the scaling behavior of both texturing methods. To this end, we generated a dataset with 10,000 images using both random textures and diffusion texturing. We then created dataset splits with various sizes and trained models for both keypoint detection and segmentation on all datasets. The performance of these models can be seen in Figure 2. For both diffusion and random textures, the performance increased with increasing dataset sizes. However, around 5000 images, the performance starts to plateau without reaching optimal performance, indicating that neither method was able to bridge the sim-to-real gap completely and obtain optimal performance. Based on this experiment, we have used a dataset size of 5000 for all other experiments in this paper.

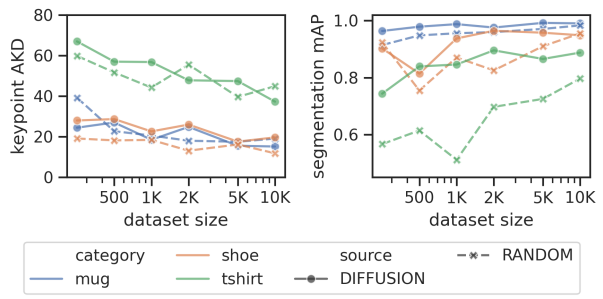

Figure 2. Scaling behavior of the different texturing approaches. For both diffusion textures and random textures, the performance improves with increasing data, though it starts to plateau around 5,000 images.

5. Discussion

In this work, we have compared text-to-image diffusion models against random textures for texturing synthetic data. We have observed that the diffusion-based texturing pipeline does not outperform random textures. This was surprising, as the diffusion-textured images appeared more realistic to us, and therefore we expected them to reduce the sim-to-real gap. We suspect that this increased realistic appearance is countered both by the tendency of the diffusion network to slightly alter the object semantics (e.g., change the shape of the mug handle slightly), polluting the annotations, and by the diffusion models leaving strong artifacts in the synthetic images on which the models can then overfit (e.g., blurring the background or smoothening transitions between objects and background). Further research is required to test these hypotheses, but there seems to be a big difference between *appearing realistic* and actually matching the distribution of real-world images.

In addition to downstream model performance, data generation speed is also important. We have not optimized this in our paper, the single-stage diffusion pipeline and random textures pipeline both took about 3 seconds to texture an image. Both can be sped up significantly and although diffusion models are becoming faster, we believe that the random textures pipeline will nonetheless be faster when fully optimized.

Finally, we note that the performance of the diffusion-based pipeline strongly depends on the context of the synthetic data. There are limits to the semantic knowledge of a diffusion model, imposed by the dataset on which it was trained. There are techniques to insert knowledge about new semantic categories [8, 30], but these come with additional engineering and data collection effort. Even for known objects, the performance can also depend on the camera angle. For example, we observed that images in which the mug handle was prominently visible tended to be more realistic than images in which the mug handle was occluded. This is in line with [22].

Overall, our diffusion-based texturing pipeline does not provide much performance gain over the random texturing approach and increases complexity. At the same time, neither method scales to achieve perfect performance, so better approaches are still needed. Improving generative models, both text-to-image and text-to-3D models, seems like the best path to reduce engineering effort in synthetic data generation, and we expect diffusion-based texturing to outperform random textures in the future. End-to-end synthetic data generation, as in [21], reduces pipeline complexity but requires methods to annotate images afterwards. For keypoints, this is even harder than for segmentation masks due to the increased precision and semantic granularity. In addition, our explicit procedure offers controllability of the generation process, allowing more control over the data distribution.

6. Conclusion

In this work, we evaluated text-to-image diffusion models for texturing synthetic images. As a testbed context, we used robotic manipulation of everyday objects. Surprisingly, our diffusion-based pipeline does not outperform texturing the 3D scenes using random textures, which is a conceptually simpler approach that is not limited to familiar objects and camera angles, unlike the diffusion pipeline. Both diffusion texturing and random textures cannot reach optimal performance, indicating that there is still a significant sim-to-real gap. We conclude that, although they remain a promising option to reduce engineering effort in synthetic data generation, the use of generative models does not provide many gains for synthetic image texturing at this time.

A. Acknowledgements

This project is funded by the Research Foundation Flanders (FWO) And the EU. Grant details will be specified upon acceptance to guarantee anonymity during the review process.

B. Research Data Availability

The codebase and data sets will be released upon acceptance.

References

[1] A survey of public datasets for computer vision tasks in precision agriculture. *Computers and Electronics in Agriculture*, 178:105760, 2020. 1

[2] Victor Besnier, Himalaya Jain, Andrei Bursuc, Matthieu Cord, and Patrick Pérez. This dataset does not exist: training models from generated images. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2020. 2

[3] Blender Online Community. Blender - a 3d modelling and rendering package. http://www.blender.org, 2024. 4

[4] Zoey Chen, Sho Kiami, Abhishek Gupta, and Vikash Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. In *Proceedings of the Robotics: Science and Systems Conference*, 2023. 1, 2, 3

[5] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 4

[6] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 4

[7] Lijie Fan, Kaifeng Chen, Dilip Krishnan, Dina Katabi, Phillip Isola, and Yonglong Tian. Scaling laws of synthetic images for model training... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7382–7392, 2024. 2, 3, 5

[8] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2023. 6

[9] Scott Geng, Cheng-Yu Hsieh, Vivek Ramanujan, Matthew Wallingford, Chun-Liang Li, Pang Wei Koh, and Ranjay Krishna. The unmet promise of synthetic training images: Using retrieved real images performs better, 2024. 2

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[11] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 4

[12] Daniel Ho, Kanishka Rao, Zhuo Xu, Eric Jang, Mohi Khansari, and Yunfei Bai. Retinagan: An object-aware approach to sim-to-real transfer. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10920–10926. IEEE, 2021. 2

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[14] Tomáš Hodaň, Vibhav Vineet, Ran Gal, Emanuel Shalev, Jon Hanzelka, Treb Connell, Pedro Urbina, Sudipta N Sinha, and Brian Guenter. Photorealistic image synthesis for object instance detection. In *2019 IEEE international conference on image processing (ICIP)*, pages 66–70. IEEE, 2019. 1, 2

[15] Joel Janai, Fatma Güney, Aseem Behl, Andreas Geiger, et al. Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Foundations and Trends® in Computer Graphics and Vision*, 12(1–3):1–308, 2020. 1

[16] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023. 4

[17] Sergey Levine, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International journal of robotics research*, 37 (4-5):421–436, 2018. 1

[18] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022. 5

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2, 4

[20] Thomas Lips, Victor-Louis De Gusseme, and Francis wyffels. Learning keypoints for robotic cloth manipulation using synthetic data. *IEEE Robotics and Automation Letters*, 9(7): 6528–6535, 2024. 1, 3, 4

[21] Chaofan Ma, Yuhuan Yang, Chen Ju, Fei Zhang, Jinxiang Liu, Yu Wang, Ya Zhang, and Yanfeng Wang. Diffusionseg: Adapting diffusion towards unsupervised object discovery. *arXiv preprint arXiv:2303.09813*, 2023. 1, 2, 3, 5, 6

[22] Wufei Ma, Qihao Liu, Jiahao Wang, Angtian Wang, Xiaoding Yuan, Yi Zhang, Zihao Xiao, Guofeng Zhang, Beijia Lu, Ruxiao Duan, Yongrui Qi, Adam Kortylewski, Yaoyao Liu, and Alan Yuille. Generating images with 3d annotations using diffusion models, 2024. 1, 2, 3, 5, 6

[23] Keith Man and Javaan Chahl. A review of synthetic image data and its use in computer vision. *Journal of Imaging*, 8 (11), 2022. 1, 2

[24] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. In *The International Symposium of Robotics Research*, pages 132–157. Springer, 2019. 1, 4

[25] Quang Nguyen, Truong Vu, Anh Tran, and Khoi Nguyen. Dataset diffusion: Diffusion-based synthetic data generation for pixel-level semantic segmentation. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3

[26] Fabian Plum, René Bulla, Hendrik K Beck, Natalie Imirzian, and David Labonte. replicant: a pipeline for generating annotated images of animals in complex environments using unreal engine. *Nature Communications*, 14(1):7195, 2023. 1, 2

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 3

[29] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017. 3

[30] Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 6

[31] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 969–977, 2018. 2

[32] Mel Vecerik, Jean-Baptiste Regli, Oleg Sushkov, David Barker, Rugile Pevceviciute, Thomas Rothörl, Raia Hadsell, Lourdes Agapito, and Jonathan Scholz. S3k: Self-supervised semantic keypoints for robotic manipulation via multi-view consistency. In *Conference on Robot Learning*, pages 449–460. PMLR, 2021. 2, 4

[33] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 4

[34] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J Cashman, and Jamie Shotton. Fake it till you make it: face analysis in the wild using synthetic data alone. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3681–3691, 2021. 1, 2

[35] Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen. Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1206–1217, 2023. 2, 3

[36] Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Jodilyn Peralta, Brian Ichter, et al. Scaling robot learning with semantically imagined experience. In *Robotics Science and Systems*, 2023. 1, 2, 3

[37] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5625–5644, 2024. 1

[38] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 1, 3

[39] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10145–10155, 2021. 2

[40] Longfei Zhou, Lin Zhang, and Nicholas Konz. Computer vision techniques in manufacturing. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 53(1):105–117, 2023. 1