Object-Centric Data Synthesis for Category-level Object Detection

Abstract

Deep learning approaches to object detection have achieved reliable detection of specific object classes in images. However, extending a model's detection capability to new object classes requires large amounts of annotated training data, which is costly and time-consuming to acquire, especially for long-tailed classes with insufficient representation in existing datasets. Here, we introduce the object-centric data setting where limited data is available in the form of multi-view images or 3D models, and we systematically evaluate the performance of four different data synthesis methods (with varying levels of contextual coherence) used to finetune object detection models on novel object categories in this setting. Unlike previous work on synthetic data generation and object compositing, which either focus on perceptual quality of synthesized images or on instance-level tasks, we specifically focus on evaluating the effectiveness of synthetic data for category-level object detection. Our experiments demonstrate significant performance gains in category-level generalization in real-world data under data-constrained regimes.

1. Introduction

Object detection is a computer vision task with widespread applications in diverse areas like healthcare, agriculture, and autonomous vehicles [3, 4, 24]. Deep-learning based object detection models require substantial amounts of annotated data to achieve robust performance in a range of visual conditions. Models trained on too little data tend to overfit and struggle to generalize to novel complex scenes [2]. For example, the creation of the COCO data set required tens of thousands of hours of manual human effort to annotate object positions and segmentation masks [33, 43]. This presents a significant time and cost barrier when scaling object detection systems to new object categories. Thus, one approach to managing data scarcity is to augment the existing real dataset with synthetic data, i.e. images that can be generated algorithmically. These images can be generated in large quantities at relatively low cost and include precise annotations by design. Moreover, synthetic generation processes can allow direct control over parameters such as object pose, lighting, occlusion, and background composition. This flexibility enables the construction of more balanced and diverse datasets, which are especially valuable when labeled real-world data is limited or costly to collect. When thoughtfully integrated, synthetic data has been shown to improve both the robustness and generalization of object detection models [40].

A common setting for object detection under data scarcity is few-shot object detection (FSOD), where one must learn to detect new object classes given only a few example images of each of these classes [26]. In this paper, we identify another setting of data scarcity where we must learn to detect novel object classes given object-centric data (*e.g.* multi-view images or 3D models of object instances). This setting is similar to practical applications where one has physical access to some instances of the novel classes (*e.g.* [46]), rather than having a small number of images of the objects placed in different scenes. In light of automatic segmentation tools (e.g. Meta's SAM-2 with video tracking [42]) or modern photogrammetry software (e.g. Epic Games' RealityScan [13]), we note that obtaining masked images and 3D models of isolated objects is feasible and relatively automatable.

We also emphasize that existing works often fail to address key considerations of practical applications, due to (**1**) insufficient modeling of occlusion or cluttered environments and (**2**) dependence on generative models, which cannot reliably synthesize long-tail object categories that are unrepresented in image datasets (e.g. specialized industrial or medical equipment), as noted by [60]. Moreover, (**3**) other approaches[11, 22, 28] have focused on optimizing the visual realism of synthesized images — including shadow generation, color harmonization, and identity preservation — evaluated through perceptual metrics such as PSNR, LPIPS, and human judgments. In contrast, our goal is to maximize the downstream utility of synthetic data for training object detection models, where factors such as viewpoint diversity and occlusion coverage may be more important than photorealism.

In this work, we develop data synthesis methods with varying levels of contextual coherence that address limitations (**1**) and (**2**), while differentiating our goals from (**3**) and focusing on the impact of each method on detection performance rather than image fidelity. We restrict ourselves

particularly to the data-limited setting where we just have access to object-centric data (complete, isolated views of objects, e.g. 3D models and multi-view images) instead of expensive annotated data of the objects placed in various environments. This paper makes the following contributions:

- **Object-centric data synthesis**: We focus on the task-specific object detection from multi-view images or 3D models of target objects and perform the first direct, controlled comparison of four data synthesis pipelines operating on object-centric inputs for category-level detection, evaluated across two model architectures ([27] and [59]) for 20K synthesized images.
- **Diffusion Copy-Paste**: We propose a novel synthesis pipeline that combines ControlNet [55] and edge conditioning with IP-Adapter [53] background conditioning to generate contextually coherent scenes while preserving exact foreground object appearances. This scheme separates background realism from object characteristics, thus avoiding the failure modes of diffusion models on long-tail or domain-specific classes [45, 60] and does not require the target classes to be represented in any pretraining dataset.
- **3D Random Placement**: We propose a 3D rendering pipeline using high dynamic range image (HDRI) environment maps that achieve physically consistent lighting without requiring depth-based plane detection or inverse rendering. This method outperforms the physically-based 3D Copy Paste method [20] especially with as few as 3000 real image examples.
- **Occlusion-stratified object detection**: We perform comprehensive stratified evaluations over six occlusion levels showing that 3D Random Placement achieves the strongest average performance across all levels, and a pairwise data-mixing strategy revealing that combining heterogeneous synthetic sources boosts performance (especially by 12% in terms of mAP for 3D Copy-Paste).

2. Related Work

2.1. Synthetic Data Generation

Synthetic data generation is a popular approach for expanding machine learning training datasets in limited-data scenarios, and much prior work has demonstrated its efficacy in different domains [19]. Popular methods for synthetic image generation include image composition using classical image processing, 3D rendering engines, and generative deep-learning approaches [36].

Traditional synthetic data methods for object detection relied on techniques such as geometric image deformations like translation and shears (*e.g.* [29]) or photometric transformations like color jittering and lighting perturbation [38]. These augmentations create visually diverse variants of the existing data, but cannot alter the underlying image scene, giving limited control over diversity of scene composition.

To synthesize novel training instances from limited data while controlling scene-level parameters such as object positioning, [12] presented the Cut-Paste pipeline. Cut-Paste relies on image composition: starting with a set of masked foreground objects, pasting them onto various background images in random orientations and applying 'blending' at the object boundaries. The blending ensures local or 'patch-level' realism in the composite images, so an object detection model trained on this data does not fixate on boundary-level discrepancies that only appear in the synthetic data. Additionally, the authors recommend using 'distractors': objects pasted onto images that do not belong to any of the target classes. These make the basic object recognition task tougher, since models can no longer identify target class objects in the image by flagging any objects that look 'out of place'. This simple approach is found to be effective in training object detection models when used alongside real annotated data.

Approaches inspired by Cut-Paste use background context to place objects in sensible positions, hence imposing global realism in the synthetic samples. Concurrent to Cut-Paste, [21] performs plane detection in indoor RGBD backgrounds to find physically plausible positions for pasting objects, and noted performance improvements over random Cut-Paste in certain settings. [11] presented a CNN 'context model' trained in self-supervised manner to suggest contextually sensible object placements. They found that random Cut-Paste [12] performs worse than using traditional data augmentation techniques, but their contextual augmentation yields significant improvement. Their experimental setting of category-level object detection differs from the instance-level setting of [12], where different objects of the same category are marked as unique labels (*e.g.* 'Cup 1', 'Cup 2') rather than being treated as the same category-level label ('Cup'). The explanation offered by [11] is that instance detection requires a model to learn fine-grain appearance of an object instance, while general object detection requires generalizing to unseen instances of the same category, which may benefit from more context modeling to provide category-relevant cues. [16] decides object placement based on local contour similarity in the background image, and they report improvements over random Cut-Paste [12] and contextual Cut-Paste [11] in the task of instance segmentation.

In the interest of reducing the synthetic-to-real domain gap, previous works have explored the use of 3D computer graphics for image generation. Rendering a complete physical scene in 3 dimensions is critical when a complete understanding of the object's position, orientation, and scale is needed, such as for 3D object detection tasks [20]. However, even for the simpler problem of detecting objects in an

image, 3D rendering can supply annotated training data that is more visually coherent, especially when using photorealistic rendering [25, 37]. For example, [47] used NVIDIA's Omniverse Isaac Sim [39] to create a pipeline for large-scale synthetic image generation using photorealistic ray-tracing, rendering randomization, and physics simulation, to achieve robust real-time instance detection.

Finally, the use of generative AI modeling for data synthesis has gained popularity in recent years. Earlier techniques using Generative Adversarial Networks (GANs) and conditional GANs enabled the creation of completely new images of target objects from a text or image prompt [6, 61]. Recently many methods have made use of diffusion models with different approaches, including generative dataset expansion guided by semantic similarity [57], or augmenting existing object detection datasets with labeled generative images [56]. Some more targeted methods include InstaGen's generation of diffusion images with annotations [18], ODGEN's image generation conditioned on provided bounding boxes annotations [60], or X-Paste's generation of foreground target objects that are then pasted onto various backgrounds [58]. The relevance of these different data synthesis techniques in our experimental setting will be elaborated in the next section.

2.2. Data-Scarce Scenarios

There exist other approaches to data-limited object detection besides synthetic data generation, such as model-oriented approaches that adapt existing object detection model architectures to the few-shot context by efficiently separating different target classes in feature space (see [34]). Though we make use of transfer learning — by fine-tuning a pretrained object detection model on data with novel object classes — we are primarily concerned with synthetic data-oriented approaches to alleviate data scarcity. These offer model-agnostic improvements which can be carried over as new and improved object detection architectures develop rapidly in the computer vision literature. These data-oriented methods also avoid inference-time overhead and require only increased training compute.

While the FSOD task assumes access to a small number of scene-level annotated images, we instead consider a setting where only object-centric data (e.g., isolated object captures or 3D models) is available. This distinction is practically important in many applications, since acquiring clean object captures is easier than collecting and annotating diverse scene images. However, this setting has been underexplored in the literature, and existing synthetic data methods are not systematically evaluated under this assumption.

Furthermore, we are concerned with the case of long-tail target classes which may not be represented in existing image datasets (such as those on which an image generation model would be trained). To elaborate on this distinction and justify our choice of data synthesis methods, we may contrast our selected methods with those found in the existing literature. For example, [58] introduce X-Paste, a modification of the Cut-Paste pipeline which uses Stable Diffusion [44] and CLIP [41] to generate and filter images of target foreground objects to be pasted onto background images. [32] applies a similar approach to the FSOD task, but still relies on pretrained diffusion models for target object generation. While these approaches improve the scalability and performance of traditional Cut-Paste by synthesizing new object examples, they are limited in their application to target classes which can reliably be generated by a diffusion model. Diffusion models struggle to reliably generate diverse images of rare objects that lack significant representation in their training set [45], and this poses a significant issue if our object class of interest is, for *e.g.*, specialized industrial or medical equipment, where errors in minor details can lead to catastrophic misclassifications. Other methods using feature extractors [35] or controllable diffusion [15] are able to retain certain visual characteristics of generated foreground objects without training on large-scale object detection datasets, but still rely on pre-trained diffusion models to fill in target object details or fail to model occlusions between objects, as noted in [60]. In contrast to the previous approaches, our proposed data augmentation technique based on conditional diffusion allows for precise control over object positions and occlusion levels; does not substantially alter the appearances of the target classes or require them to be represented in existing datasets; and can still condition background generation on an image prompt (see also [30]).

2.3. Image Compositing Approaches

Finally, our work is different from object compositing approaches such as ORIDa [28], ObjectDrop [51], Paint-by-Example [52], and ObjectStitch [49] although both their approaches and ours utilize common techniques such as Cut-Paste composition and diffusion-based image synthesis. These works tackle a generative task by synthesizing photorealistic images with correct shadows, color harmonization, and identity preservation and use image similarity metrics such as PSNR, LPIPS, DINO and CLIP to evaluate human perceptual quality. In contrast, our approach tackles a discriminative task by focusing on object detection and using the mAP@50 performance metric to evaluate downstream object detection capability.

3. Data Synthesis Methods

We select data synthesis methods which require either a set of masked images of target objects (for 2D Cut-Paste or Diffusion Copy-Paste) which can be generated using automatic segmentation tools like SAM2 [42], or 3D models of target

and distractor objects (for 3D Random Placement and 3D Copy-Paste) which can be generated with automatic photogrammetry tools (*e.g.* [13]). Moreover, our chosen methods cover a diverse range of image processing and generation techniques and incorporate varying degrees of contextual coherency of target objects in the synthesized images. We are particularly interested in understanding how incorporating synthetic images derived from object-centric data can improve object detection in our experimental settings.

3.1. Cut-Paste

As described in Sec. 2.1, the Cut-Paste method [12] is a straightforward yet effective approach to synthetic data generation. The process involves extracting object instances from existing images using segmentation masks and then compositing them onto new background scenes (see Fig. 1). To ensure variety, the method includes augmentations such as rotation, scaling, occlusion, truncation, and different blending strategies. These additions help create diverse training examples that are both realistic and challenging for object detectors.

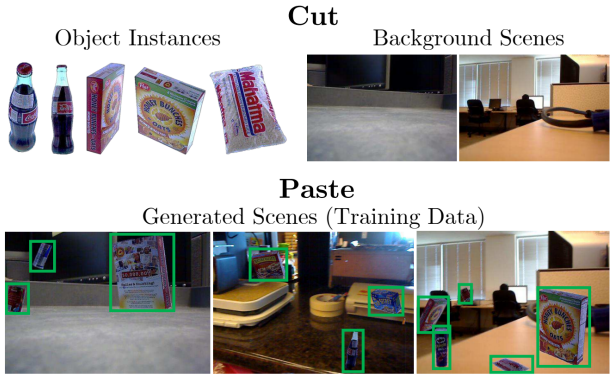

Figure 1. Examples from the Cut-Paste implementation of [12].

One of the key benefits of this method is its ability to produce high-coverage datasets with minimal human intervention. Since object masks and backgrounds can be reused in multiple combinations, the approach significantly increases visual diversity across backgrounds and scales. The method also bypasses the need for precise scene layout or scene lighting information, which are often difficult to replicate or transfer between domains. As a result, it becomes easier to build datasets tailored to specific target environments or rare object instances. We adapt the original implementation (made available by the authors at `https://github.com/debidatta/syndata-generation`), porting from Python 2 to 3, adding proportional object scaling, improving the occlusion checks by using object masks, and switching the Poisson Blur effect to OpenCV's Seamless Cloning [7] for speed.

3.2. Diffusion Copy-Paste

We propose a novel method using generative models for synthetic data generation. This approach, termed *Diffusion Copy-Paste*, uses diffusion models conditioned on certain edge and image prompts. In situations where it is important to maintain the visual appearance of target objects found in training data (such as for long-tailed or sensitive target classes), one can control the distribution of generated images using conditional diffusion tools. For example, we use ControlNet [55], a neural network architecture for adding spatial conditioning controls (in our case, edge map conditioning) to large, pretrained text-to-image diffusion models. Additionally, we use an Image Prompt Adapter [53], which conditions the background of the synthesized image to match a reference background image. Given pictures of the environment one wants to perform object detection in, Diffusion Copy-Paste can use these by passing them in to the Image Prompt Adapter, hence creating images which have similar backgrounds to the target environment.

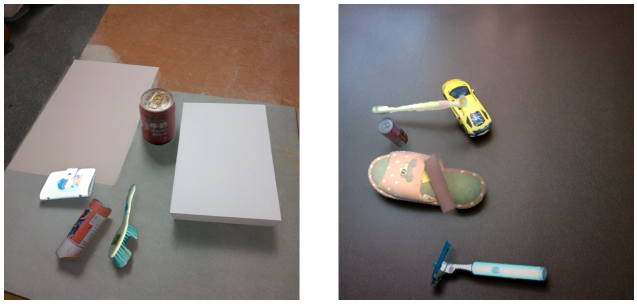

Figure 2. Some examples of the Diffusion Copy-Paste image generation technique.

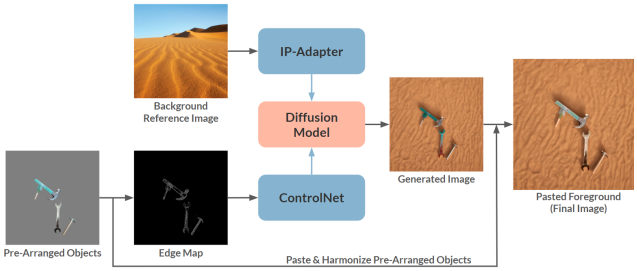

Figure 3. Pipeline for synthetic data generation for Diffusion Copy-Paste.

Fig. 3 demonstrates the complete Diffusion Copy-Paste pipeline: **1)** pre-arranged objects (obtained by running Cut-Paste with a blank background) are passed into the Canny edge detector algorithm [8] to obtain an edge map that encodes their positions and outlines within the picture. **2)** we generate a synthetic image conditioned to follow a reference background image prompt (through IP-Adapter) and

have edges matching the object layout edge map (through ControlNet). Since the foreground objects generated by the diffusion model are often nonsensical, we then paste the original pre-arranged objects and perform basic image harmonization to obtain the final image.

The implementation of the diffusion method expands on the 2D Cut & Paste method but also accesses certain core and custom nodes from the ComfyUI backbone [9] for running the image generation model and image post-processing tasks. The specific image generation model used was RealVisXL v5.0 Lightning [14], a diffusion model optimized for fast photorealistic image generation.

3.3. 3D Random Placement

We also propose a method for utilizing the 3D geometry of the objects by rendering 3D scenes with objects randomly placed in them. This method is referred to as *3D Random Placement (3D RP)* and employs HDRI as backgrounds. HDRI scenes capture complete 360° views of real environments while preserving lighting intensity on each image pixel. This property ensures the 3D scene has consistent and realistic lighting without the need for added light sources, thereby allowing foreground objects to blend in naturally with the background (as seen in Fig. 4). In this approach, objects are still inserted randomly into the background, leading to objects that float in mid-air and have no contextual relation to the background.

Figure 4. From HDRI background to synthetic data.

We arrange multiple 3D scenes in the Blender rendering engine [5] (using Python bindings), each with multiple target & distractor objects placed in different positions. 3D RP renders each scene multiple times with varying camera angles, zoom levels, and exposure levels, to capture diverse object viewpoints (as seen from Fig. 5) and model the objects' appearances under various lighting conditions.

Figure 5. A single can viewed from multiple angles.

3.4. 3D Copy-Paste

Adapted from [20], this method adds contextual coherence to the 3D Random Placement method by placing objects in a collision-free manner on feasible horizontal planes, which are detected within the RGBD background images using clustering algorithms. The two primary improvements of 3D Copy-Paste over 3D Random Placement are physically plausible object placements (using plane detection) and lighting estimation (using deep inverse rendering techniques [31]).

The original implementation utilizes two primary computational steps: geometric analysis of scenes and spatially-varying illumination estimation. Agglomerative Hierarchical Clustering [17] identifies planar surfaces from RGBD point clouds, and the 'floor' plane is chosen as the lowest horizontal plane. The scene's lighting is estimated using deep inverse rendering through LightNet [31], which predicts environment maps of omnidirectional light distribution over discrete 4×4 pixel areas.

Our adaptation of 3D Copy-Paste involved five new features: (1) allowing objects to be inserted on multiple horizontal surfaces instead of just the floor plane. (2) inserting multiple objects in a scene (we sequentially search for free space and insert objects one-by-one). (3) realistic object sizing (we detect and prevent scenarios where objects take up too much space in the camera's view). (4) rotating objects along the x, y, z axes, rather than z-axis only (we remove the assumption that objects have a 'base' which always rests on the floor). (5) 3D collision detection (this replaces the original 2D approximations with precise 3D bounding box collision checks). Figure 6 summarizes our improvements.

Figure 6. Our improvements on the original 3D Copy-Paste method [20].

3.5. Comparison

The four data synthesis methods vary in the input data types they utilize and they incorporate different amounts of visual context into the images they synthesize. While the objects in 2D Cut-Paste and 3D Random Placement have almost no relation to the backgrounds, the Diffusion Copy-Paste and 3D Copy-Paste methods use contextual information about the foreground objects and backgrounds to produce more coherent and realistic object placements. 3D rendering of objects and incorporation of background context comes at a

computational cost, however, with 2D Cut-Paste producing synthetic images at about 10 times the rate (25 imgs/sec) as the other three methods, though all of the synthetic data methods are still far cheaper to execute than real annotated data collection. Figure 7 contains example images generated by these methods and an overview of their benefits and drawbacks.

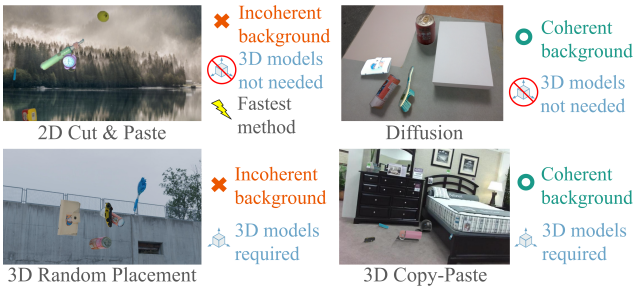

Figure 7. Qualitative comparison of our proposed data generation methods.

4. Evaluation Dataset

To evaluate our different methods of generating synthetic data, we would like to train object detection models on these datasets and evaluate these models on real-world data. As described in the previous sections, we are particularly interested in situations where labeled training data containing target objects may not be available in large amounts, but object-centric data is available to us. In light of this, we select the PACE dataset [54] as our source of real-world evaluation data. PACE consists of 300 videos with $55,000$ frames and $258,000$ object annotations, covering 238 object instances from 43 categories in cluttered scenes. The authors also provide 3D textured models for each object instance. The dataset is accompanied by PACE-Sim, which uses BlenderProc's physically-based rendering [10] to create $100,000$ synthetic images with 2.4 million annotations across 931 object instances, but we discard this since it was generated using similar methods to 3D RP and 3D Copy-Paste (which use the Blender engine), making it unsuitable for the purpose of evaluation.

Out of the 43 object categories represented in PACE, 13 are either represented in COCO [33] or closely related to classes in COCO (*e.g.* PACE has mugs while COCO has cups). We discard these classes since our experiments use the YOLO11 [27] and RT-DETR [59] object detection models, which were pretrained on the COCO dataset, hence giving them out-of-the-box detection capabilities on these. Out of the remaining 30 classes, we discard objects with fewer than 10 unique instances appearing in the real videos, and choose our final object classes to be "toy car", "can", "snack box", and "ramen box".

To set up an experimental setting which tests generalization across different instances of the same category, we split the instances for each class into training, validation, and test splits in a $2:1:2$ ratio, while ensuring that instances that co-occur in any video are allocated to the same split (this is done by creating an instance co-occurrence graph and using a greedy algorithm to split the components in the desired ratio). This allows us to then assign each video to the same split as all the object instances that appear in the video, resulting in the split described in Table 1.

	Train	Val	Test
Toy Car Instances	5	2	4
Can Instances	4	2	4
Snack Box Instances	7	4	7
Ramen Box Instances	6	3	5
Total Instances	22	11	20
Videos	75	33	75
Frames	11,498	5,547	10,825

Table 1. Statistics of our dataset across train, validation, and test splits, including per-class object instances.

5. Experiments and Results

5.1. Design Choices

We generate 20k images for each synthetic data method, using the object-centric data from the PACE dataset. Cut-Paste and Diffusion Copy-Paste use foreground cutouts for the target and distractor instances found in video frames of the training data. As backgrounds, they use PACE video frames that lie completely outside the train-val-test sets above, to avoid test leakage or target objects appearing in the background without being assigned labels. 3D RP and 3D Copy-Paste use 3D models for the same train instances of target and distractors, and they use external datasets for the backgrounds, namely HDRI images from PolyHaven [23] and RGBD images from SUN-RGBD [48].

We choose the YOLO11 object detection model [27] for our experiments due to its real-time performance [1] and the actively-maintained Ultralytics library. YOLO11 was trained on the COCO dataset [33] with 80 object classes, but we apply transfer learning by replacing the last layer to classify objects between the 4 categories chosen in our PACE dataset.

5.2. Sequential versus Mixed schemes

We first test two different schemes for training the model. In the *sequential* scheme, we first finetune the model on a synthetic dataset only, and then finetune on a fraction of the available PACE training data. In the *mixed* scheme, we

combine a synthetic dataset with a fraction of the PACE train data. The sequential and mixed data schemes model the practical scenario where some real annotated data is available (in limited amounts), and synthetic data is used to augment the real data. In the mixed scheme real & synthetic data are naively combined, while the sequential scheme first finetunes on the abundant synthetic data (to approximately learn the target data distribution) and then finetunes on the limited real data which matches the target data distribution. Previous work [50] found the finetuning scheme outperforms the mixed scheme regardless of real data quantity, and we find similar results in our experimental setup, as seen in Table 2.

	2.5%	5%	10%
Cut-Paste	55.6 / 34.8	56.1 / 46.2	56.3 / 51.1
Diffusion CP	60.5 / 52.1	60.2 / 55.3	58.4 / 55.0
3D RP	**62.8** / 51.7	**64.0** / 52.9	**63.2** / 56.9
3D CP	58.2 / 59.8	59.6 / 53.5	54.7 / 59.7

Table 2. YOLO11 finetuning results (test set mAP@50) with different methods and varying proportions of real data. Inner entries are in the form *Sequential* / *Mixed* performance.

5.3. Head-to-Head comparison

As described in Section 3.5, the different data synthesis methods differ in their incorporation of contextual information and in data generation speed. Thus, to directly compare the performance scaling across methods as we increase the number of training images, we performed a head-to-head comparison in which an off-the-shelf YOLO11 model was finetuned on the same number of images of each data type. The results are presented in Table 3, and indicate the strong performance of our Diffusion Copy-Paste and 3D Random Placement methods as standalone sources of data, only being surpassed in the scenario where one has 10,000 real images to train on (which exists outside the limited-data regime).

5.4. Performance across models

We compare the performance of YOLO11 using the sequential training paradigm (Fig. 8) against that of another object detection model, RT-DETR [59] (Fig. 9), to investigate if our performance gains hold across different model architectures[1]. We observe the same significant mAP boost from our data augmentation methods, again with strong performances from both 3D Random Placement and Diffusion Copy-Paste. For either model we observe that performance from sequential and real-only training schemes eventually converge (when 10,000 real images are available for training) but the performance gap between them is substantial in the low-data regime. This shows that a model finetuned on synthetic data offers significant improvement over a pretrained model, when followed by finetuning on a small amount of real data.

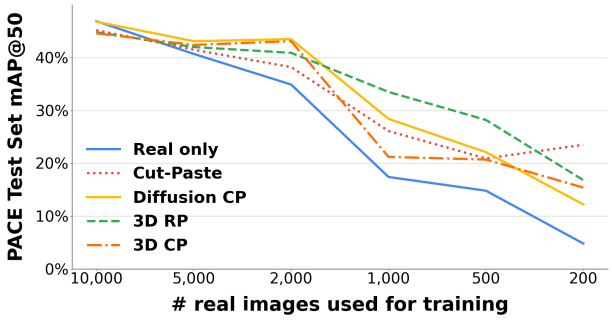

Figure 8. Sequential training of YOLO11 on different amounts of real data, compared to only training on real data.

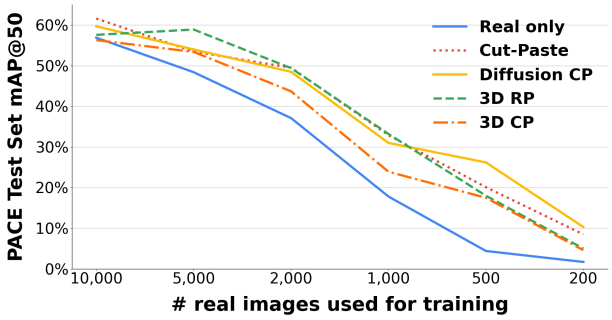

Figure 9. Sequential training of RT-DETR on different amounts of real data, compared to only training on real data.

5.5. Performance across occlusion levels

We investigate the ability of our data synthesis methods to aid in detecting occluded objects in Fig. 10. The occlusion percentage, calculated as the total overlap area divided by the total area of all foreground objects, is binned into six categories, ranging from 0% to roughly 25% occlusion. We find that our 3D Random Placement method achieves the best results on average across all bins by this metric, followed by the strong performance from the Cut-Paste method.

5.6. Data Mixing

We also examined the effect of different data mixing schemes, including combining multiple types of synthetic data to increase dataset diversity. As expected, the runs mixed with the real data performed the best, with a dominant performance from the Diffusion Copy-Paste method

[1]We do not perform hyperparameter optimization in this experiment due to the lack of a consistent validation set when varying the availability of real images.

	200	500	1,000	2,000	5,000	10,000	20,000
Real only	0.04	0.15	0.17	0.35	0.41	**0.47**	
Cut-Paste	0.35	0.34	0.34	0.32	0.38	0.39	0.33
Diffusion CP	**0.41**	**0.38**	**0.42**	0.39	**0.44**	0.41	0.37
3D RP	0.38	0.38	0.40	**0.45**	0.42	0.43	**0.43**
3D CP	0.30	0.37	0.40	0.27	0.33	0.37	0.30

Table 3. Head-to-head comparison of YOLO11 performance (test set mAP@50) with the four different data synthesis methods. The off-the-shelf model is finetuned on the number of images specified by the column, of type specified by the row.

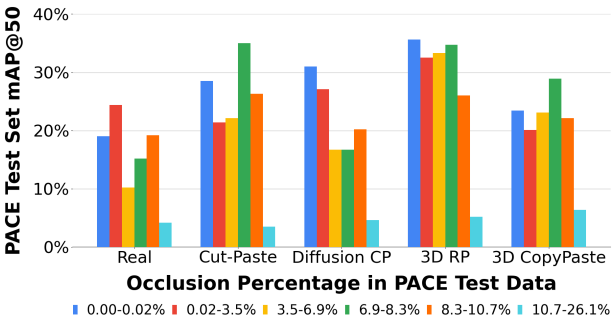

Figure 10. PACE test set performance stratified by occlusion percentage (i.e., the total overlap area divided by the total area of all foreground objects).

(see Table 4). Aside from these runs, however, we observed that mixing a synthetic data type with another synthetic method improved performance by an average of 3.2% mAP for Cut-Paste, 0.36% mAP for 3D Random Placement, 3.57% mAP for Diffusion Copy-Paste, and 12.1% mAP for 3D Copy-Paste. This suggests that combining multiple data augmentation methods can yield substantial performance boosts and compensate for deficiencies from any particular method.

Mix with	Real	C-P	3DRP	DiffCP	3DCP
Real	37.4%	34.9%	42.2%	50.4%	44.5%
C-P	–	34.9%	41.9%	36.9%	35.5%
3DRP	–	–	43.6%	45.7%	44.3%
DiffCP	–	–	–	39.5%	46.6%
3DCP	–	–	–	–	30.0%

Table 4. mAP results for models trained on individual datasets (diagonal) and pairwise combinations (off-diagonal). Here C-P stands for Cut-Paste, 3DRP for 3D Random Placement, DiffCP for Diffusion Copy-Paste, and 3DCP for 3D Copy-Paste.

6. Discussion

In this paper, we discussed different methods of generating synthetic data for improving object detection in data-limited settings. We compared methods using both 2D and 3D object information, as well as methods with random versus contextually guided object placement. Our specific experimental setup involving data scarcity and occlusion-modeling, both motivated by constraints arising from a practical application of object detection, leads us to apply two data synthesis methods from the literature (Cut-Paste and 3D Copy-Paste) and implement two of our own (Diffusion Copy-Paste and 3D Random Placement) tailored to these challenges. Our results demonstrate the considerable power of synthetic data augmentation in this data-limited regime, and particularly showed that our developed methods achieve strong performance boosts over widely-used and relevant data synthesis methods in the literature, such as Cut-Paste.

Comparison across four methods reveals mixed results for when modeling visual context and realism can lead to improved performance. 3D Random Placement and Diffusion Copy-Paste produce more realistic datasets than Cut-Paste by incorporating geometric, lighting, & perspective information, and contextual background information respectively, and both outperform the Cut-Paste baseline. 3D Copy-Paste combines aspects of both but fails to surpass Cut-Paste. We note that adding certain *realistic* constraints (e.g. placing objects on surfaces) can unintentionally *reduce diversity* along other aspects, such as viewpoint coverage, which may counteract the advantages of modeling the real world more faithfully.

References

[1] Momina Liaqat Ali and Zhou Zhang. The yolo framework: A comprehensive review of evolution, applications, and benchmarks in object detection. *Computers*, 13(12), 2024. 6

[2] Simone Antonelli, Danilo Avola, Luigi Cinque, Donato Crisostomi, Gian Luca Foresti, Fabio Galasso, Marco Raoul Marini, Alessio Mecca, and Daniele Pannone. Few-shot object detection: A survey. *ACM Computing Surveys (CSUR)*, 54(11s):1–37, 2022. 1

[3] Chetan M Badgujar, Alwin Poulose, and Hao Gan. Agricultural object detection with you only look once (yolo) algorithm: A bibliometric and systematic literature review. *Computers and Electronics in Agriculture*, 223:109090, 2024. 1

[4] Abhishek Balasubramaniam and Sudeep Pasricha. Object detection in autonomous vehicles: Status and open challenges, 2022. 1

[5] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. 5

[6] Christopher Bowles, Liang Chen, Ricardo Guerrero, Paul Bentley, Roger Gunn, Alexander Hammers, David Alexander Dickie, Maria Valdés Hernández, Joanna Wardlaw, and Daniel Rueckert. GAN augmentation: Augmenting training data using generative adversarial networks. *arXiv preprint arXiv:1810.10863*, 2018. 3

[7] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 4

[8] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986. 4

[9] Comfy Org. ComfyUI Official Documentation. https://docs.comfy.org/, 2024. 5

[10] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Knauer, Klaus H. Strobl, Matthias Humt, and Rudolph Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023. 6

[11] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *Proceedings of the european conference on computer vision (ECCV)*, pages 364–380, 2018. 1, 2

[12] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection . In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1310–1319, Los Alamitos, CA, USA, 2017. IEEE Computer Society. 2, 4

[13] Epic Games, Inc. RealityScan. http://www.realityscan.com, 2025. 1, 4

[14] Evgeny. Realvisxl v5.0 lightning. https://huggingface.co/SG161222/RealVisXL_V5.0_Lightning, 2024. Accessed: 2025-09-15. 5

[15] Haoyang Fang, Boran Han, Shuai Zhang, Su Zhou, Cuixiong Hu, and Wen-Ming Ye. Data augmentation for object detection via controllable diffusion models. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1257–1266, 2024. 3

[16] Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 682–691, 2019. 2

[17] Chen Feng, Yuichi Taguchi, and Vineet R. Kamat. Fast plane extraction in organized point clouds using agglomerative hierarchical clustering. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6218–6225, 2014. 5

[18] Chengjian Feng, Yujie Zhong, Zequn Jie, Weidi Xie, and Lin Ma. InstaGen: Enhancing object detection by training on synthetic dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14121–14130, 2024. 3

[19] Alvaro Figueira and Bruno Vaz. Survey on synthetic data generation, evaluation methods and gans. *Mathematics*, 10 (15), 2022. 2

[20] Yunhao Ge, Hong-Xing Yu, Cheng Zhao, Yuliang Guo, Xinyu Huang, Liu Ren, Laurent Itti, and Jiajun Wu. 3d copy-paste: Physically plausible object insertion for monocular 3d detection. In *Advances in Neural Information Processing Systems*, pages 17057–17071. Curran Associates, Inc., 2023. 2, 5

[21] Georgios Georgakis, Arsalan Mousavian, Alexander C. Berg, and Jana Kosecka. Synthesizing training data for object detection in indoor scenes, 2017. 2

[22] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *2021 CVPR Proceedings*, pages 2918–2928, 2021. 1

[23] Zaal Greg, Tuytel Rob, Majboroda Sergej, Savva Dimitrios, and Guest Jarod. Poly haven: The public 3d asset library. https://polyhaven.com/hdris. 6

[24] Weijie He, Yuwei Zhang, Ting Xu, Tai An, Yingbin Liang, and Bo Zhang. *Object Detection for Medical Image Analysis: Insights from the RT-DETR Model*, page 415–420. Association for Computing Machinery, New York, NY, USA, 2025. 1

[25] Tomás Hodan, Vibhav Vineet, Ran Gal, Emanuel Shalev, Jon Hanzelka, Treb Connell, Pedro Urbina, Sudipta N. Sinha, and Brian K. Guenter. Photorealistic image synthesis for object instance detection. *IEEE International Conference on Image Processing (ICIP)*, pages 66–70, 2019. 3

[26] Gabriel Huang, Issam Laradji, David Vazquez, Simon Lacoste-Julien, and Pau Rodriguez. A survey of self-supervised and few-shot object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4071–4089, 2022. 1

[27] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. Ultralytics YOLO, 2023. 2, 6

[28] Jinwoo Kim, Sangmin Han, Jinho Jeong, Jiwoo Choi, Dongyeoung Kim, and Seon Joo Kim. Orida: Object-centric real-world image composition dataset. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3051–3060, 2025. 1, 3

[29] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002. 2

[30] Yu Li, Xingyu Qiu, Yuqian Fu, Jie Chen, Tianwen Qian, Xu Zheng, Danda Pani Paudel, Yanwei Fu, Xuanjing Huang, Luc Van Gool, et al. Domain-rag: Retrieval-guided compositional image generation for cross-domain few-shot object detection. *arXiv preprint arXiv:2506.05872*, 2025. 3

[31] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2475–2484, 2020. 5

[32] Shaobo Lin, Kun Wang, Xingyu Zeng, and Rui Zhao. Explore the power of synthetic data on few-shot object detection. In *Proceedings of the IEEE/CVF conference on com-*

puter vision and pattern recognition, pages 638–647, 2023. 3

[33] 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*, pages 740–755, Cham, 2014. Springer International Publishing. 1, 6

[34] Tianying Liu, Lu Zhang, Yang Wang, Jihong Guan, Yanwei Fu, Jiajia Zhao, and Shuigeng Zhou. Recent few-shot object detection algorithms: A survey with performance comparison. *ACM Transactions on Intelligent Systems and Technology*, 14(4):1–36, 2023. 3

[35] Yanxing Liu, Jiancheng Pan, and Bingchen Zhang. Control Copy-Paste: Controllable Diffusion-Based Augmentation Method for Remote Sensing Few-Shot Object Detection. *arXiv preprint arXiv:2507.21816*, 2025. 3

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

[37] Yair Movshovitz-Attias, Takeo Kanade, and Yaser Sheikh. How useful is photo-realistic rendering for visual learning? In *Computer Vision – ECCV 2016 Workshops*, pages 202–217, Cham, 2016. Springer International Publishing. 3

[38] Alhassan Mumuni, Fuseini Mumuni, and Nana Kobina Gerrar. A survey of synthetic data augmentation methods in machine vision. *Machine Intelligence Research*, 21(5):831–869, 2024. 2

[39] NVIDIA Corporation. Isaac sim simulator documentation. https://docs.isaacsim.omniverse.nvidia.com/4.5.0/index.html. 3

[40] Enes Özeren and Arka Bhowmick. Evaluating the impact of synthetic data on object detection tasks in autonomous driving. *arXiv preprint arXiv:2503.09803*, 2025. 1

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 3

[42] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1, 3

[43] Tal Remez, Jonathan Huang, and Matthew Brown. Learning to segment via cut-and-paste. *arXiv preprint arXiv:1803.06414*, 2018. 1

[44] 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. 3

[45] Dvir Samuel, Rami Ben-Ari, Simon Raviv, Nir Darshan, and Gal Chechik. Generating images of rare concepts using pre-trained diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4695–4703, 2024. 2, 3

[46] Faranak Shamsafar, Sunil Jaiswal, Benjamin Kelkel, Kireeti Bodduna, and Klaus Illgner-Fehns. Leveraging multi-view data for improved detection performance: An industrial use case. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4464–4471, 2023. 1

[47] Ritvik Singh, Jingzhou Liu, Karl Van Wyk, Yu-Wei Chao, Jean-Francois Lafleche, Florian Shkurti, Nathan Ratliff, and Ankur Handa. Synthetica: Large scale synthetic data for robot perception. *arXiv preprint arXiv:2410.21153*, 2024. 3

[48] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 6

[49] Yizhi Song, Zhifei Zhang, Zhe Lin, Scott Cohen, Brian Price, Jianming Zhang, Soo Ye Kim, and Daniel Aliaga. Object-stitch: Object compositing with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18310–18319, 2023. 3

[50] Bram Vanherle, Steven Moonen, Frank Van Reeth, and Nick Michiels. Analysis of training object detection models with synthetic data. *arXiv preprint arXiv:2211.16066*, 2022. 7

[51] Daniel Winter, Matan Cohen, Shlomi Fruchter, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. Objectdrop: Bootstrapping counterfactuals for photorealistic object removal and insertion. In *European Conference on Computer Vision*, pages 112–129. Springer, 2024. 3

[52] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18381–18391, 2023. 3

[53] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 2, 4

[54] Yang You, Kai Xiong, Zhening Yang, Zhengxiang Huang, Junwei Zhou, Ruoxi Shi, Zhou Fang, Adam W Harley, Leonidas Guibas, and Cewu Lu. PACE: A large-scale dataset with pose annotations in cluttered environments. In *ECCV 2024*, pages 473–489. Springer, 2024. 6

[55] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, pages 3813–3824, 2023. 2, 4

[56] Manlin Zhang, Jie Wu, Yuxi Ren, Jiahong Yang, Ming Li, and Andy J Ma. DiffusionEngine: Diffusion model is scalable data engine for object detection. *Pattern Recognition*, page 112141, 2025. 3

[57] Yifan Zhang, Daquan Zhou, Bryan Hooi, Kai Wang, and Jiashi Feng. Expanding small-scale datasets with guided imagination. *Advances in neural information processing systems*, 36:76558–76618, 2023. 3

[58] Hanqing Zhao, Dianmo Sheng, Jianmin Bao, Dongdong Chen, Dong Chen, Fang Wen, Lu Yuan, Ce Liu, Wenbo Zhou, Qi Chu, et al. X-Paste: Revisiting scalable copy-

paste for instance segmentation using CLIP and StableDiffusion. In *International Conference on Machine Learning*, pages 42098–42109. PMLR, 2023. 3

[59] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16965–16974, 2024. 2, 6, 7

[60] Jingyuan Zhu, Shiyu Li, Yuxuan Andy Liu, Jian Yuan, Ping Huang, Jiulong Shan, and Huimin Ma. ODGEN: Domain-specific object detection data generation with diffusion models. *Advances in Neural Information Processing Systems*, 37: 63599–63633, 2024. 1, 2, 3

[61] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 3