The effect of PV array instances and images backgrounds on OOD generalization

Gabriel Kasmi*†, Laurent Dubus*, Philippe Blanc†, Yves-Marie Saint Drenan†

November 29, 2021

Abstract

Despite remarkable performances, convolutional neural networks (CNNs) can experience large performance drop when used on new, previously unseen data. This problem of domain shift limits the ability of such methods to be used in practical settings. Recent work (Gulrajani and Lopez-Paz, 2020) showed that the minimization of the empirical risk (ERM, Vapnik (1998)), which consists in minimizing average of the loss function over the training dataset is a strong baseline for addressing domain shift. Thus, the problem of domain shift can be rephrased as a problem of out-of-distribution (OOD) generalization since unseen domains boil down in this case to unseen data drawn from distributions distinct from that of the training dataset. Improving OOD performance rests on understanding the failure modes of ERM. A common framework for that is to isolate how ERM relies on both relevant and spurious or background features to make its predictions. To the best of our knowledge, no work has so far focused on the identification of such relevant/spurious features in practical settings.

We focus on remote PV detection. Our goal is to disentangle factors that cause a poor OOD performance, as reported by Wang et al. (2017). We consider a synthetic dataset and try to quantify the impact of the type of array (array instance) and the type of background on OOD performance. We show that (1) the performance drop is mostly caused by new backgrounds and that (2) the type of background in the training dataset has a decisive importance on OOD generalisation.

Besides we wonder it is possible to quantify the importance of these semantic labels in the latent representation of the model. To do so, we use the method based on mutual information recently proposed by Islam et al. (2021). Our assumption is that depending on whether confronted to a "hard" or "simple" case, the network will encode more or less background features. To test this assumption, we first wonder how much background and relevant features does a CNN encode in its latent representation. We then wonder whether this amount varies across visual classes and backgrounds.

1 Introduction

Deep learning based models for remote sensing of solar arrays often experience an impredictible performance drop when deployed to a new location Wang et al. (2017). This problem is caused by the fact that machine learning methods struggle to generalize out-of-domain (OOD). In this

[1]Réseau de Transport d'électricité
[2]Ecole nationale supérieure des Mines de Paris

poster, we design an experimental setup that aims at disentangling the impact of the background and the solar array type on OOD performance.
The setup consists in a synthetic dataset that mixes different types of solar arrays (which we call "instances") and different types of background. We leverage this synthetic dataset to study OOD generalization in two directions : - In the first case, we consider a fixed source dataset and see whether a model fails to generalize to new instances or new backgrounds - In the second case, we consider a fixed target dataset and see whether OOD generalization can be affected by the composition of the source dataset
In order to provide insights on the uneven ability to generalize, we leverage Islam et al. (2021) dimensionality estimation technique to see whether depending on the instance and background, the number of dimensions in the latent that encode the semantic factors "solar array" and "background" varies.
The questions we wish to address are the following :

- Is the failure to generalize predominantly due to unseen arrays or unseen backgrounds ?
- Has the type of background or array instance an influence on OOD performance ?
- Can we quantify which types of backgrounds or solar arrays are better for generalization ?

2 Related literature

PV array detection The literature on PV array detection using machine learning methods has been flourishing over the last few years. First works relied on hand-crafted features (Malof et al., 2015, 2016) but quickly deep learning models (especially CNNs) have become the *de facto* mainstream method due to their unprecedented performance and has enabled solar array mapping over increasingly larger geographical areas (Yu et al., 2018; Malof et al., 2019; Mayer et al., 2020). Recent work focus on the automated construction of PV registries (Rausch et al., 2020) and the improvement of in-domain performance, i.e. more accurate segmentation masks and the reduction of false positives in the detections. To this end, Jie et al. (2020) proposed to tweak the network architecture in order to provide more accurate segmentation masks. However, as highlighted by De Jong et al. (2020) or de Hoog et al. (2020) despite their performance, methods still lack reliability when deployed to unseen areas.

Out-of-distribution (OOD) generalization OOD generalization refers to the ability of a model to perform well on examples that are outside of the training distribution. It has been widely documented that models can perform poorly when tested outside of their training distributions (Torralba and Efros, 2011). Yet, in many real-world application, testing distributions that are different from the training distribution are the rule rather and the exception. In the case of PV panel detection, OOD generalization happens when new images (no matter more recent, with different characteristics or from another geographical area) that are not in the training distribution are given to the model. We also talk in this setting about geographical domain shift (Huang et al., 2020). As such, it is necessary to design models that are have better OOD generalization abilities since the cost of annotating data is very expensive and therefore limits the ability to use models in practical settings.
As recently reviewed by Gulrajani and Lopez-Paz (2020), numerous methods have been proposed to improve the domain generalization ability of the models. Yet in their experiments, they showed

that the empirical risk minimization (Vapnik, 1998) constitutes a very strong baseline. As such, in order to improve the OOD generalization of the model, we need to better understand the failures of ERM. Several studies (Beery et al., 2018; Nagarajan et al., 2020) have illustrated how ERM fails in characteristic ways. One of such ways is that the ERM relies on both *predictive* and *spurious* features to fit the training data. While predictive features are indeed relevant, spurious features are in general dataset-dependent (Gulrajani and Lopez-Paz, 2020) and will therefore mislead the model when deployed to new settings.

Assessing what machines learn Understanding the inner workings of classification models is crucial to improve their OOD generalization ability. There exist a wide stream of literature, often related to explainable AI (XAI), aiming at providing tools that allow researchers and practitioners to understand the inner workings of complex models such as CNNs (see for instance the review of Lucieri et al. (2020)). One key concern is that a model will make good prediction for wrong reasons, which implies to base outcomes on spurious correlations (Lapuschkin et al., 2019). This requires to unveil what the algorithm has learned. In order to do that, visualisation techniques have been developed, such as class activation maps (Zhou et al., 2016) as well as feature disentanglement techniques (Esser et al., 2020). Curation techniques leveraging the distinction between predictive and spurious features have been proposed (Puli et al., 2021). In the latter example, the proposed model has been tested on chest X-ray, a domain where the distinction between background noise (related to the sensor and the place where the images are taken) and the predictive features is also at play.
Feature disentanglement techniques are a promising way to quantify the amount of a given information that is encoded in the representation of the model. Islam et al. (2021) recently leveraged the framework proposed by Esser et al. (2020) to quantify the amount of "shape" or "texture" information that is encoded in the latent representation of modern CNNs, thus documenting the "texture bias" highlighted by Hermann and Lampinen (2020). In order to identify predictive and spurious features, we propose to use the framework as Islam et al. (2021).

3 Method

3.1 Synthetic dataset

The synthetic dataset consists in two domains. A source domain, comprised of 80,000 samples with 4 array types and 2 background types and a target domain comprised of 4 array types (different from those of the source domain) and one background type. Each domain is splitted into training, validation and testing datasets. We also include to intermediate domain testing datasets, one containing source backgrounds and target arrays, and the other source arrays and the target background.
These images come from IGN aerial images, that are accessible here and provided under open license.Source arrays are depicted on figure 1 and some example images are provided on figure 2.
The dataset is balanced, for one positive sample, one negative sample is generated. Each subgroup of data is also evenly represented. More precisely, the source domain includes 8 (array, instances) pairs and for each pair, we generate the same number of samples (positive and negatives). Moreover, we generate label files for each of the subgroups, as well as for group of subgroups in order to be able to train the model on subsamples of the training dataset only. See table 1 for a detailed depiction of the characteristics of the source domain.

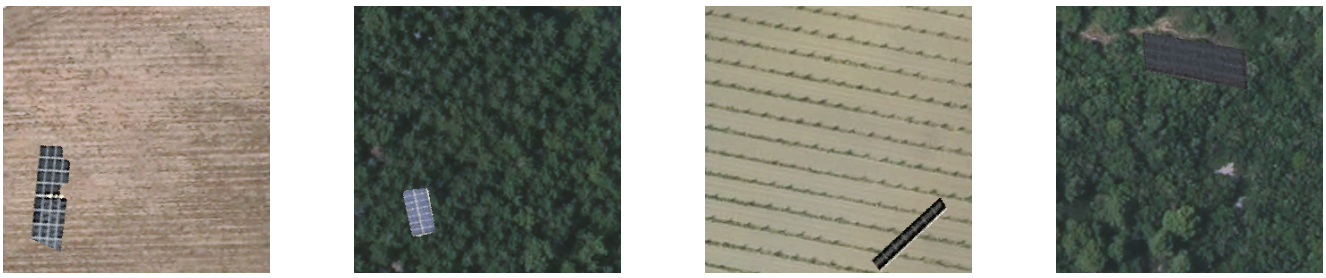
The target domain includes 4 (array, instances) pairs and for each pair, we generate the same number of samples. Intermediate domains are also balanced in terms of positive samples and type of arrays and backgrounds.
Arrays are randomly applied on a background image, with a random horizontal and vertical flips and a random rotation. Arrays are also randomly shifted from their initial position.

Array \ Location	Forests	Mountains	Overall
Black *(Small/Large)*	10000 *(5000/5000)*	10000 *(5000/5000)*	20000 *(10000/10000)*
Blue *(Small/Large)*	10000 *(5000/5000)*	10000 *(5000/5000)*	20000 *(10000/10000)*
Total *(Small/Large)*	20000 *(10000/10000)*	20000 *(10000/10000)*	40000 *(20000/20000)*

Table 1: Number of positively labelled images for each category. The dataset contains 40000 additional negatively labelled background images

(a) Large blue array (b) Small blue array (c) Large black array (d) Small black array

Figure 1: Arrays replicated to construct the dataset



(a) Fields (with a large blue array) (b) Forest (with a small blue array) (c) Field (with a small black array) (d) Forest (with a large black array)

Figure 2: Examples of data samples from the experimental dataset

3.2 Dimensionality estimation

Disentangling factors in a latent representation The starting point is the method proposed by Esser et al. (2020) for explaining latent representation. The idea is to map the latent representation of an input image, denoted $z \in \mathbb{R}^N$ into factors $(\tilde{z}_k)_{k=0}^K$ such that $\forall k \in \{0, \ldots, K\},\ \tilde{z}_k \in \mathbb{R}^{N_k}$

and $\sum_{k=0}^{K} N_k = N$. This mapping is done using an invertible neural network (IIN) T. Each factor is then interpreted as a given feature (e.g. animal species, shape, texture, color). Since the concepts we are interested in are not exhaustive, the factor $\tilde{z}_0$ is introduced as a residual that captures the remaining variability, which is not identified by the factors in $\{1, \ldots, K\}$. Semantic concepts are identified through their dimensionality: the most complex are assumed to have a larger dimensionality over the most simple ones.
It is assumed that the factors are independent and follow a Gaussian distribution. The factors are identified using the similarity a given concept between two images x^a and x^b which given our assumptions translates into a positive mutual information between the latent representation of the input images. The dimensionality N_F of the semantic concept F is then derived from a score s_F. This score corresponds to the sum of the correlation coefficient between each component in the latent representation :

$$s_F = \sum_{i=1}^{N} \frac{Cov(z_i^a, z_i^b)}{\sqrt{Var(z_i^a)Var(z_i^b)}} \in [-N, N]$$

where z_i denothes the ith component of z. The dimensionality of the residual factor then ensures that the dimensionality N_F of the Fth semantic factor is positive. We refer the reader to Esser et al. (2020) for more technical details.

Identifying the semantic concepts in practice We re-use the script written by Islam et al. (2021) for their estimation of the shape and texture on images in order to avoid potential caveats. The only different is that we feed our own images to the dimensionality estimation module.
In order to estimate a semantic factor
The method of Esser et al. (2020) requires several training pairs (x^a, x^b) in order to compute T. In order to meet the data requirements, Islam et al. (2021) used style transfer in order to define image pairs that are similar in texture but not in shape (so that the semantic concept of shape is identified) and other pairs that consist in identical images with different textures.
In the case of our source domain dataset, the dimensionality of a given concept is estimated by considering two samples that share nothing but this semantic concept. For instance, in order to estimate the dimensionality of the semantic concept "array", since there are two backgrounds we will sample an image of array on the forest background and another from the fields background. In order to estimate the dimensionality of the semantic concept "fields", we consider two images that depict the fields, but with a varying foreground (which can be either another instance of array or no array).

Robustness checks Robustness checks are provided in the appendix A.

3.3 Model and training

Our baseline model is a ResNet50 with random initialization. As it can be seen from the appendix A.1, the estimated dimentionalities are not model dependent. We use data augmentation during training. Models are trained during 30 epochs. Model is early stopped if there is no improvement over the validation dataset for more than 5 epochs.

4 Results

4.1 OOD performance

We first decompose the out-of-domain error into two components:

- The error cause by the fact that the model has to identify solar array instances that it has never seen before,
- The error caused by the fact that the model has to identify solar arrays over unseen backgrounds.

To isolate these effects, we evaluate OOD performance in three settings, depicted on figure 3

- We consider new solar array instances but in domain backgrounds (lefmost boxplot of figure 3)
- In-domain solar array instances but new backgroun (boxplot in the middle of figure 3)
- Out-of-domain array instances and backgrounds (rightmost boxtplot of figure 3)

We can see that the change in backgrounds drives the OOD error. Put otherwise, according to our experiment, if a model fails to generalize well out-of-domain, it is mostly due to the fact that out-of-domain samples depict unseen backgrounds. A possible explanation for this phenomenon is that when facing small objects such as solar arrays, detection models will heavily rely on background features to make their prediction. However, as recalled by Gulrajani and Lopez-Paz (2020) or Nagarajan et. al. (2020), features extracted from the background of the image are spurious features in the sense that they are likely to change when one is shifted from one domain to the other.
In order to further inspect the impact of the background and the type of solar array instance on OOD performance, we perform a second experiment where the target dataset remains fixed and the composition of the training dataset changes. Our hope is to show some backgrounds and some solar array instances can allow for a better OOD generalization than others.

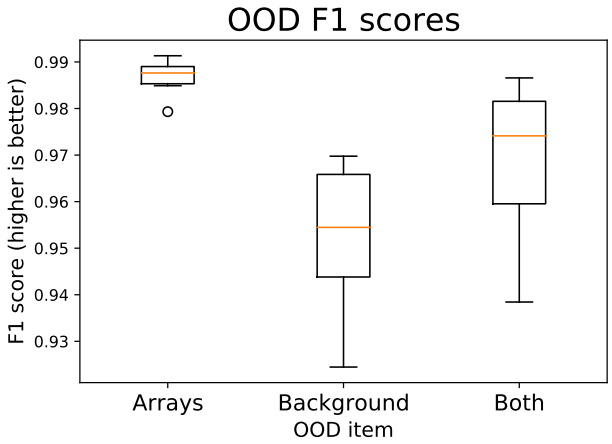


Figure 3: OOD F1 scores for a fixed training dataset.

4.2 The impact of the source domain on OOD performance

We now consider the reverse phenomenon and set a fixed OOD dataset with unseen array instances and an unseen background. The composition of the training dataset on the other hand varies : it contains more or less large or blue arrays (y-axis) and more or less images drawn over the fields background (x-axis). Each cell outputs the average F1 score of the model on the OOD dataset, given a fixed share of (background, array instance) in the training dataset.
We can see on figure 4 that the composition of the training dataset has an important impact on performance. Moreover, the final performance is more affected by the background than by the solar array instance. This conforts the idea according to which some background characteristics prevent the model from learning too many spurious features during training. In our case, a plausible explanation is that arrays are more contrasted on fields backgrounds than on forest backgrounds and therefore making the distinction between the background (which is irrelevant) and the foreground (i.e. the solar array) more explicit.

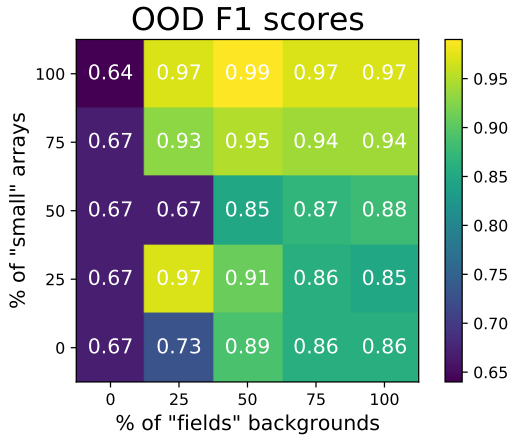


Figure 4: F1 OOD score for a given (array,background) share in the training dataset.

4.3 Dimensionality estimates

As it can be seen from figure 5, our results are inconclusive. All instances of solar arrays have the same estimated dimensionality (around 400) and all types of backgrounds also have the same dimensionality estimation (around 400 as well). As such, based on these results, it is not possible to say that there is a correlation between the dimensionality of the instance and how it is suited for OOD generalisation.

5 Conclusion

This experiment shows that for small object detection on overhead imagery, OOD performance is mostly affected by the background characteristics. A possible explanation is that some backgrounds allows for a better disentanglement between predictive (i.e. correlated with the semantic label one wants to predict) and spurious (i.e. correlated with the training dataset) features.
Future work should therefore forcus on consolidating this claim in a more principled framework. To this end, it is necessary to take into account additional factors that can vary from one dataset

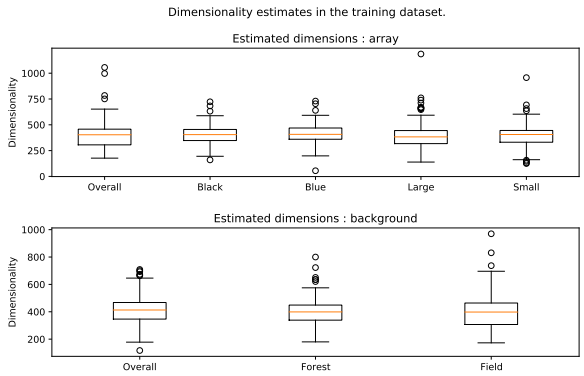


Figure 5: Dimensionality estimates of instances of the semantic concepts

to another such as the image characteristics (ground sampling distance, brightness, projection of the ground on the image). It is also necessary to show that on "good" backgrounds, the model does indeed extract predictive features.

References

Beery, S., Van Horn, G., and Perona, P. (2018). Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473. 3

de Hoog, J., Maetschke, S., Ilfrich, P., and Kolluri, R. R. (2020). Using satellite and aerial imagery for identification of solar pv: State of the art and research opportunities. In *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*, pages 308–313. 2

De Jong, T., Bromuri, S., Chang, X., Debusschere, M., Rosenski, N., Schartner, C., Strauch, K., Boehmer, M., and Curier, L. (2020). Monitoring spatial sustainable development: semi-automated analysis of satellite and aerial images for energy transition and sustainability indicators. *arXiv preprint arXiv:2009.05738*. 2

Esser, P., Rombach, R., and Ommer, B. (2020). A disentangling invertible interpretation network for explaining latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9223–9232. 3, 4, 5

Gulrajani, I. and Lopez-Paz, D. (2020). In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*. 1, 2, 3

Hermann, K. L. and Lampinen, A. K. (2020). What shapes feature representations? exploring datasets, architectures, and training. *arXiv preprint arXiv:2006.12433*. 3

Huang, B., Bradbury, K., Collins, L. M., and Malof, J. M. (2020). Do deep learning models generalize to overhead imagery from novel geographic domains? the xgd benchmark problem. In *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 1476–1479. IEEE. 2

Islam, M. A., Kowal, M., Esser, P., Jia, S., Ommer, B., Derpanis, K. G., and Bruce, N. (2021). Shape or texture: Understanding discriminative features in cnns. *arXiv preprint arXiv:2101.11604*. 1, 2, 3, 5

Jie, Y., Yue, A., Liu, S., Huang, Q., Chen, J., Meng, Y., Deng, Y., and Yu, Z. (2020). Photovoltaic power station identification using refined encoder–decoder network with channel attention and chained residual dilated convolutions. *Journal of Applied Remote Sensing*, 14(1):016506. 2

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., and Müller, K.-R. (2019). Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1–8. 3

Lucieri, A., Bajwa, M. N., Dengel, A., and Ahmed, S. (2020). Achievements and challenges in explaining deep learning based computer-aided diagnosis systems. *arXiv preprint arXiv:2011.13169*. 3

Malof, J. M., Bradbury, K., Collins, L. M., and Newell, R. G. (2016). Automatic detection of solar photovoltaic arrays in high resolution aerial imagery. *Applied energy*, 183:229–240. 2

Malof, J. M., Hou, R., Collins, L. M., Bradbury, K., and Newell, R. (2015). Automatic solar photovoltaic panel detection in satellite imagery. In *2015 International Conference on Renewable Energy Research and Applications (ICRERA)*, pages 1428–1431. IEEE. 2

Malof, J. M., Li, B., Huang, B., Bradbury, K., and Stretslov, A. (2019). Mapping solar array location, size, and capacity using deep learning and overhead imagery. *arXiv preprint arXiv:1902.10895*. 2

Mayer, K., Wang, Z., Arlt, M.-L., Neumann, D., and Rajagopal, R. (2020). Deepsolar for germany: A deep learning framework for pv system mapping from aerial imagery. In *2020 International Conference on Smart Energy Systems and Technologies (SEST)*, pages 1–6. IEEE. 2

Nagarajan, V., Andreassen, A., and Neyshabur, B. (2020). Understanding the failure modes of out-of-distribution generalization. *arXiv preprint arXiv:2010.15775*. 3

Puli, A., Zhang, L. H., Oermann, E. K., and Ranganath, R. (2021). Predictive modeling in the presence of nuisance-induced spurious correlations. *arXiv preprint arXiv:2107.00520*. 3

Rausch, B., Mayer, K., Arlt, M.-L., Gust, G., Staudt, P., Weinhardt, C., Neumann, D., and Rajagopal, R. (2020). An enriched automated pv registry: Combining image recognition and 3d building data. *arXiv preprint arXiv:2012.03690*. 2, 11, 12

Torralba, A. and Efros, A. A. (2011). Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE. 2

Vapnik, V. (1998). Statistical learning theory. *Wiley series on adaptive and learning systems for signal processing, communications and control*. 1, 3

Wang, R., Camilo, J., Collins, L. M., Bradbury, K., and Malof, J. M. (2017). The poor generalization of deep convolutional networks to aerial imagery from new geographic locations: an empirical study with solar array detection. In *2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–8. IEEE. 1

Yu, J., Wang, Z., Majumdar, A., and Rajagopal, R. (2018). Deepsolar: A machine learning framework to efficiently construct a solar deployment database in the united states. *Joule*, 2(12):2605–2617. 2

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929. 3

A Robustness checks

A.1 Consistency of the estimated dimensionalities

Model \ Factor	**Array**	**Background**	**Residual**
Inception	408 *[169, 646]*	410 *[195, 624]*	1230 *[953, 1507]*
Resnet pretrained	415 *[179, 651]*	396 *[180, 609]*	1238 *[1005, 1473]*
Resnet random	405 *[117, 694]*	413 *[169, 657]*	1230 *[993, 1467]*

Table 2: Dimensionality estimates for the factors array, background and the residual. Brackets display the 95% confidence interval. All models have been trained over the whole dataset. Inception refers to the model developped by Rausch et al. (2020) and fine tuned on our dataset. Resnet pretrained refers to a Resnet50 model pretrained on ImageNet and Resnet Rand refers to a Resnet50 model with a random initialization. Confidence intervals are obtained after 100 bootstrap iterations on the test set to compute the representations and estimate the dimensionality of the factors.

Model \ Factor	**Array**	**Residual**
Inception	525 *[372, 678]*	1523 *[1370, 1676]*
Resnet pretrained	542 *[375, 709]*	1519 *[1339, 1673]*
Resnet random	529 *[323, 735]*	1518 *[1313, 1724]*

Table 3: Dimensionality estimates for the factors array and the residual. Brackets display the 95% confidence interval. All models have been trained over the whole dataset. Inception refers to the model developped by Rausch et al. (2020) and fine tuned on our dataset. Resnet pretrained refers to a Resnet50 model pretrained on ImageNet and Resnet Rand refers to a Resnet50 model with a random initialization. Confidence intervals are obtained after 100 bootstrap iterations on the test set to compute the representations and estimate the dimensionality of the factors.

Factor / Model	Background	Residual
Inception	513 *[295, 731]*	1534 *[1317, 1753]*
Resnet pretrained	544 *[292, 796]*	1504 *[1252, 1756]*
Resnet random	524 *[290, 758]*	1523 *[1290, 1757]*

Table 4: Dimensionality estimates for the factors background and the residual. Brackets display the 95% confidence interval. All models have been trained over the whole dataset. Inception refers to the model developped by Rausch et al. (2020) and fine tuned on our dataset. Resnet pretrained refers to a Resnet50 model pretrained on ImageNet and Resnet Rand refers to a Resnet50 model with a random initialization. Confidence intervals are obtained after 100 bootstrap iterations on the test set to compute the representations and estimate the dimensionality of the factors.

A.2 Reality check

In this check, we see whether there is a discrepancy between the estimated dimensions in the synthetic dataset and in real images. It turns out that the estimation of the semantic concepts arrays and backgrounds are about the same magnitude with real images than with synthetic data.

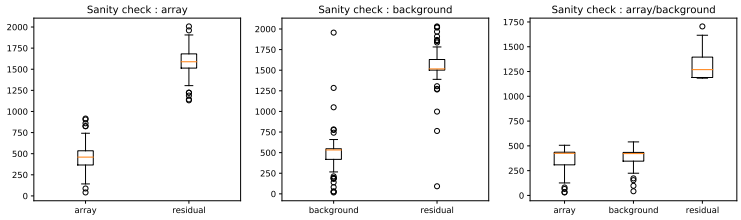


Figure 6: Reality check

A.3 Random check

In this check, we randomly match images to see whether the estimates significantly differ from those that we would obtain randomly. It turns out that it is the case.

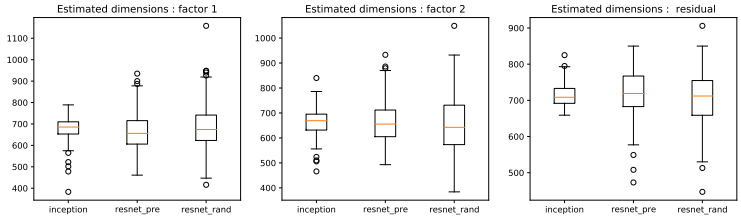


Figure 7: Random check