CANDLE: An Image Dataset for Causal Analysis in Disentangled Representations

Abbavaram Gowtham Reddy
IIT Hyderabad, India
cs19resch11002@iith.ac.in

Benin Godfrey L
IIT Hyderabad, India
benin.godfrey@cse.iith.ac.in

Vineeth N Balasubramanian
IIT Hyderabad, India
vineethnb@iith.ac.in

1. Appendix

In this appendix, we provide the following details.

- Comparison of CANDLE with existing datasets
- Extensibility of CANDLE dataset and ground-truth metadata structure
- Observed confounding on CANDLE dataset
- Some definitions of causality used in this work
- Visual Explanation of UC, CG metrics
- Analysis of UC metric
- Algorithms for implementation of UC, CG metrics
- Time Complexity of UC, CG Metrics
- Additional Experiments
- Counterfactual images generated while computing CG metric
- Additional qualitative results of experiments on CANDLE and synthetic datasets

2. Comparison of CANDLE With Existing Datasets

We compare CANDLE with existing datasets in various aspects. Table 1 and Figure 1 provide more the details. Clearly CANDLE combines various desiderata of ideal and more complex datasets.

3. Extensibility of CANDLE

Since CANDLE is a simulated rendering of 3D primitives with properties placed in a HDRI background, the dataset itself is easy to extend adding different variations of each of them, rerendering a different version suitable for some specific downstream task that requires it. As such, care is taken to ensure that extensibility is one of the goals that this dataset satisfies implicitly. Each factor of variation is placed in a separate folder with every variation being a separate '.blend' file containing one instance of that variation. The dataset generating script then iterates through every file picking up labels of each variation from the filenames. This makes it simple to add, remove and replace variations with minimal knowledge of 3D graphics or Blender. Some points to note for compatibility are:

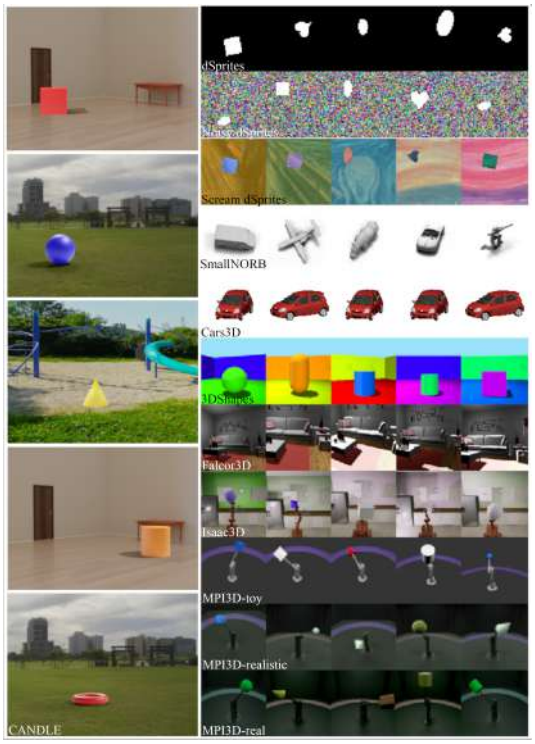

Figure 1: Comparison of sample images from various datasets. Datasets *(Left)*: CANDLE *(Right: from top to bottom)*: dSprites, Noisy dSprites, Scream dSprites, SmallNORB, Cars3D, 3DShapes, Falcor3D, Isaac3D, MPI3D-toy, MPI3D-realistic, MPI3D-real. Note that CANDLE is the only dataset with a realistic scene and a foreground object controlled by several latent factors among all these datasets.

- Objects are placed at the origin of the 3D space with the primitive scaled down to occupy 1x1x1 m of space. This allows scaling to affect all objects similarly and maintain semantic consistency with the backgrounds.
- Scenes with different HDRIs can by made by modifying the world texture to pick up a different image from the filesystem as its surface color.
- For Materials and Lights, just adding a .blend file with only the property suffices.

Dataset	Number of Factors	Image Resolution	3D	Realistic	Presence of Foreground Object	Foreground Object not Centered	Complex Background	Confounders
dSprites	5	64×64	✗	✗	✓	✓	✗	✗
Noisy dsprites	5	64×64	✗	✗	✓	✓	✗	✗
Scream dsprites	5	64×64	✗	✗	✓	✓	✗	✗
SmallNORB	5	128×128	✓	✗	✓	✗	✗	✗
Cars3D	3	64×64	✓	✗	✓	✗	✗	✗
3Dshapes	6	64×64	✓	✗	✓	✗	✗	✗
Falcor3D	7	128×128	✓	✗	✗	✗	✓	✗
Isaac3D	9	128×128	✓	✗	✓	✓	✓	✗
MPI3D-toy	7	64×64	✓	✗	✓	✓	✗	✗
MPI3D-realistic	7	256×256	✓	✓	✓	✓	✗	✗
MPI3D-real	7	512×512	✓	✓	✓	✓	✗	✗
CANDLE	6	320×240	✓	✓	✓	✓	✓	✓

Table 1: Comparison of various existing datasets with CANDLE. CANDLE stands out after comparing with existing datasets along various dimensions.

- The filename and the property in the .blend file must match for proper querying. For example, for adding pink color, call the material "Pink" in Blender and name the file "`pink.blend`".

Though these details might not provide value to direct end-users of the dataset, this is one of the benefits of simulating a dataset rather than capturing one, especially if it can provide realistic renditions - which can be leveraged by certain interested users of the dataset.

4. Metadata of CANDLE

The below JSON file contains metadata corresponding to image number 123. Size takes three values - small: 1, medium: 1.5, large: 2. These JSON files can be used to get different weak supervisions like rank pairing, match pairing and restricted labeling [1] along with bounding boxes of foreground objects.

```
{"123":{
    "scene": "indoor",
    "lights": "left",
    "objects":{"cube_0":{
            "object_type":"cube",
            "color":"red",
            "size":1.5,
            "rotation":60,
            "bounds":[[81, 56],[120, 88
                ]]}}
}}
```

5. Observed Confounding on CANDLE

We can conditionally select images from the dataset to mimic observed confounding as depicted in Figure 2. For example, we can select images where cubes appear in red and spheres appear in green. Confounding can be applied to any subset of nodes. For example, we can select images where certain objects only appear in certain scenes. This allows our dataset to be used in ways beyond shown in this work too.

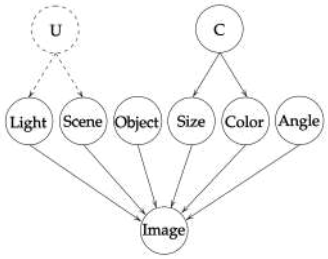

Figure 2: Observed confounding(C) can be realised from the full dataset by conditionally selecting images.

6. Example Counterfactual Questions

Counterfactual image generation has many uses including counterfactual data augmentation, fairness analysis [3] etc. Let us see an example on how CANDLE can be used in the study of counterfactual generation. Consider Figure 3, where the left column contains images we are given, and the right column contains counterfactual images we are interested in. For example, in row 1, given a red sphere, we would like to know "how the image would have looked like if the color of the sphere had been yellow?", a counterfactual. in row 2, given a red sphere, our question would be "how the

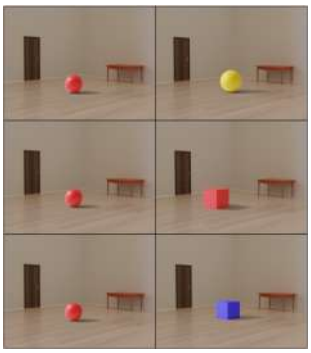

Figure 3: Visual demonstration of some counterfactual questions

image would have looked like if the shape of the object had been cube?". In row 3, our question would be "how the image would have looked like if we change the shape to cube and color to blue?". Since we have such ground-truth available, we can perform evaluation of methods that learn underlying structure and generate counterfactual images using this dataset.

7. Definitions

Definition 7.1. (Average Causal Effect). The *Average Causal Effect(ACE)* of a random variable X on a random variable Y for a particular treatment $do(X = \alpha)$ with reference to a baseline treatment $do(X = baseline(X))$ is defined as $ACE^{Y}_{do(X=\alpha)} = \mathbb{E}[Y|do(X = \alpha)] - \mathbb{E}[Y|do(X = baseline(X))]$

Definition 7.2. (Individual Causal Effect). The *Individual Causal Effect(ICE)* of a random variable X on a random variable Y for a particular treatment $X = \alpha$ with reference to a baseline treatment $do(X = baseline(X))$ is defined as $ICE^{Y}_{do(X=\alpha)} = P[Y|do(X = \alpha)] - P[Y|do(X = baseline(X)]$

Definition 7.3. (Disentangled Causal Process). When generative factors G do not causally influence each other but can be confounded by a set of confounders C, a causal model for Y with generative factors G is said to be disentangled if and only if the structural equation for Y is a function of only generative factors G and noise variable N_Y.

8. Visual Explanation of UC and CG metrics

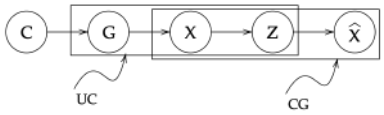

Figure 4: Relationship among various components of causal process and UC, CG metrics. UC relates G, X, and Z, CG relates X, Z and $\hat{X}$.

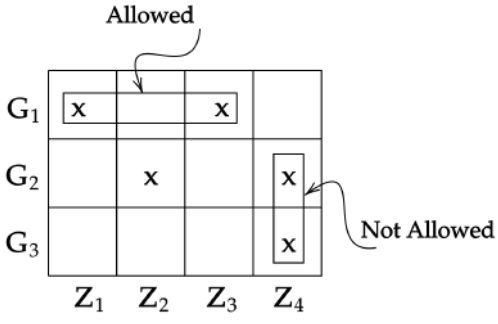

Figure 5: Visual explanation of UC metric. According to UC metric, it is allowed G_1 to be captured by Z_1, Z_3 but it is not allowed for Z_4 to capture both G_2, G_3.

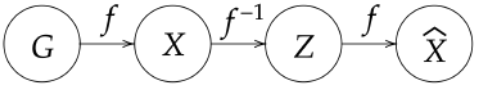

Figure 6: Generative factors G generate observations X through some function f, our goal in learning disentanglement representation is to learn f^{-1} and f that transform observations into latent factors Z and latent factors back to observations $\hat{X}$. For a perfect disentanglement and generation, $X = \hat{X}$.

9. Analysis of UC Metric

UC metric produces results that are densely distributed near 1 because of the way Jaccard similarity behaves. This can be seen with the help of the following example. Consider the case where we have two generative factors ($N = 2$), 4 latent dimensions ($M = 4$) and $\rho = 3$. Now, let the latents corresponding to the two generative factors be $\{1, 2, 3\}$ and $\{2, 3, 4\}$ respectively. UC measure gives the value of 0.5 even though there is a significant overlap in two sets. One solution is to use some transformation on the UC score to make the scores evenly distributed in the range [0,1]. For example, $e^{2(x-1)}$ is a good choice to be a transformation function on UC, where x is UC score calculated. This problem arises only when we choose to map more than one latent dimension to each generative factor. This effect of densely distributed UC scores around 1 is not a problem when we compare methods on UC metric because relative values are important in comparison, not the absolute values. Transformation on UC measure is useful when we attribute more than one latent dimension to generative factors. Please note that in the experiments section of main paper, we have used original version of UC equation without any transformation as described here as we are presenting results only when $\rho = 1, \rho = 2$.

10. Algorithms for UC, CG Metrics

The implementation steps to compute the UC and CG metrics are shown below in Algorithms 1 and 2 respectively.

Algorithm 1: Unconfoundedness (UC) Metric

Inputs: generative factors G, latent dimensions Z, IRS function;
Result: UC metric
$UC = 0; k = |G|$;
for $i = 0;\ i < k;\ i++$ **do**
  Z_I = IRS(G_i);
  for $j = i + 1;\ i < k - 1;\ j++$ **do**
    Z_J = IRS(G_j);
    $UC = UC + \frac{|Z_I \cap Z_J|}{|Z_I \cup Z_J|}$;
  end
end
$UC = 1 - \frac{UC}{k*(k-1)}$;
return UC;

Algorithm 2: Counterfactual Generativeness (CG) Metric

Inputs: generative factors G, latent dimensions Z, IRS function, dataset $\mathbb{D}$;
Result: CG metric
$CG = 0$; $k = |G|$; $n = |\mathbb{D}|$;
for $i = 0;\ i < k;\ i++$ **do**
  Z_I = IRS(G_i);
end
$ACE = 0$;
for $i = 0;\ i < n;\ i++$ **do**
  $x = x_i$;
  $x_I^{cf1} = decoder(x|do(Z_I = Z_I^x))$;
  $x_I^{cf2} = decoder(x|do(Z_I = baseline(Z_I))$;
  $ICE_{Z_I^x}^{x_I^{cf}} = |P(G_{ik}|x_I^{cf1}) - P(G_{ik}|x_I^{cf2})|$;
  $x_{\setminus I}^{cf1} = decoder(x|do(Z_{\setminus I} = Z_{\setminus I}^x))$;
  $x_{\setminus I}^{cf2} = decoder(x|do(Z_{\setminus I} I = baseline(Z_{\setminus I}))$;
  $ICE_{Z_{\setminus I}^x}^{x_I^{cf}} = |P(G_{ik}|x_{\setminus I}^{cf1}) - P(G_{ik}|x_{\setminus I}^{cf2})|$;
  $ACE = ACE + |ICE_{Z_I^x}^{x_I^{cf}} - ICE_{Z_{\setminus I}^x}^{x_I^{cf}}|$;
end
$CG = \frac{ACE}{n}$;
return CG;

11. Time Complexity of UC, CG Metrics

In UC measure, we are learning about representative latents Z_I for G_i using the IRS method [2], which was proved to run in $\mathcal{O}(|D|)$ [2], where $|D|$ is the size of dataset. Once we obtain Z_I corresponding to G_i, we need to evaluate the UC measure which takes $\mathcal{O}(|G|^2)$. Where $|G|$ is the number of generative factors which is usually a small number. To evaluate CG, we need to evaluate the prediction probabilities of G_{ik} given some generated image $\hat{x}$. Since the classifier is pre-trained. We can evaluate CG simply using two forward passes through the network for each generative factor which runs in $\mathcal{O}(|D| \times |G|)$ with respect to neural network forward propagation time. As $|G|$ is usually a small number, practical time complexity of both UC and CG metrics is approximately linear in time w.r.t. input dataset size.

12. Additional Experiments

Synthetic Dataset: We created a synthetic fully confounded dataset as shown in Figure 7 to understand existing models behaviour under extreme confounding. Reconstructions and latent traversal of β-VAE model trained on the synthetic dataset reveal that both color and shape are confounded by a set of latents in the generative model. Whenever we change latents related to color, shape also changes along with color(see Appendix for visual illustrations). We expect UC measure to give a score of 0, which it does. Even though the latents corresponding to a generative factors are obtained using IRS matrix [2], the IRS score is close to 1(Table 2).

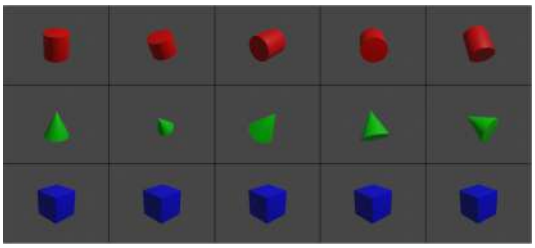

Figure 7: Synthetic dataset with extreme confounding effect where cylinders appear in red, cones appear in green, cubes appear in blue.

Model	IRS	UC $\rho = 1$	CG $\rho = 1$
β-VAE	0.99	0	0.01
β-TCVAE	0.99	0	0.04
DIP-VAE	0.99	0	0.03
Factor-VAE	0.99	0	0.04

Table 2: Comparison of IRS, UC and CG metrics on synthetic dataset for various models.

This shows that the IRS metric is not suitable for measuring the degree of unconfounding achieved by a model. Even though there are two generative factors, due to extreme confounding, the models are combining semantic concepts and learning latents such that color and shape are treated as a single factor(e.g. models treat inputs as toys: red cylinders, green cones, blue cubes) which is expected from existing models. Accuracy of the trained model used in CG metric is close to 100%

MPI3D-Toy Dataset: We performed experiments on MPI3D-Toy dataset without any observable confounding, where semi supervised methods are getting a UC score of 1 when $\rho = 1$. But when $\rho = 2$, the results show the limitations of existing models in disentangling completely. All methods fail to reconstruct images well which shows the limitations of reconstruction capabilities of existing latent variable models. Accuracy of the trained model used in CG metric is 98.34%

Model	IRS	UC $\rho = 1$	CG $\rho = 1$	UC $\rho = 2$	CG $\rho = 2$
β-VAE	0.53	0.52	**0.23**	0.19	**0.24**
β-TCVAE	0.57	0.28	0.22	0.19	0.14
DIP-VAE	0.27	0.28	0.10	0.19	0.14
Factor-VAE	0.58	0.52	0.14	0.34	0.16
SS-β-VAE	0.60	**1.00**	0.10	**0.66**	0.09
SS-β-TCVAE	**0.64**	**1.00**	0.09	**0.66**	0.15
SS-DIP-VAE	0.35	0.42	0.10	0.19	0.11
SS-Factor-VAE	0.56	**1.00**	0.12	0.60	0.14

Table 3: Comparison of IRS, UC and CG metrics on MPI3D-toy dataset for various models.

Comparison with DCI Metric Since UC metric is similar to Disentanglement (D) of DCI metric(represented as $DCI(D)$ in Table 5), we compare different methods agaiist the DCI metric along with IRS, UC and CG in the next set of experiments. DCI metric uses linear regressors to quantify the importance of latent dimensions in predicting generative factors but causal influences cannot be identified using regressors. Since UC metric uses IRS score to get the latent dimensions corresponding to generative factors, causal dimensions are being used in calculating UC score.

Model	IRS	DCI (D)	UC $\rho=1$	CG $\rho=1$	UC $\rho=2$	CG $\rho=2$
β-VAE	0.72	0.11	0.40	0.16	0.27	0.17
β-TCVAE	0.74	0.14	0.40	0.22	0.26	0.21
DIP-VAE	**0.75**	0.09	0.40	0.18	0.53	0.13
Factor-VAE	0.71	0.12	0.40	0.20	0.26	0.20
SS-β-VAE	0.51	**0.20**	**0.90**	0.20	0.27	0.22
SS-β-TCVAE	0.46	0.16	0.40	**0.24**	0.26	**0.26**
SS-DIP-VAE	0.28	0.18	0.7	0.15	**0.60**	0.24
SS-Factor-VAE	0.45	0.19	0.4	0.15	0.26	0.15

Table 4: Comparison of DCI, IRS, UC, CG metrics on CANDLE dataset for various models when experimented with full dataset without any observable confounding

Model	IRS	DCI (D)	UC $\rho=1$	CG $\rho=1$	UC $\rho=2$	CG $\rho=2$
β-VAE	**0.80**	0.17	0.40	0.21	0.13	0.20
β-TCVAE	0.50	0.10	0.70	0.20	0.29	0.20
DIP-VAE	0.24	0.03	0.40	0.13	0.29	0.14
Factor-VAE	0.47	0.13	0.40	0.22	0.13	0.15
SS-β-VAE	0.51	0.18	**0.80**	0.28	**0.60**	0.19
SS-β-TCVAE	0.49	0.19	**0.80**	0.25	0.46	**0.25**
SS-DIP-VAE	0.29	0.17	0.40	0.32	0.26	0.19
SS-Factor-VAE	0.46	**0.21**	0.40	**0.36**	**0.60**	0.21

Table 5: Comparison of DCI, IRS, UC and CG metrics on CANDLE dataset for various models when experimented with confounding dataset where where certain shapes appear in certain colors as given in Table 6

Object	Cube	Sphere	Cylinder	Cone	Torus
Available Colors	Red Blue	Blue Yellow	Yellow Purple	Purple Orange	Orange Red

Table 6: Chosen confounding between object and color for experiments provided in Table 5

Model	IRS	DCI (D)	UC $\rho=1$	CG $\rho=1$	UC $\rho=2$	CG $\rho=2$
β-VAE	0.75	0.17	0.40	0.20	0.27	0.20
β-TCVAE	**0.81**	0.13	0.40	0.11	0.26	0.11
DIP-VAE	0.76	0.10	0.70	0.19	0.46	0.17
Factor-VAE	0.74	0.11	0.70	**0.21**	0.47	0.21
SS-β-VAE	0.53	0.21	**0.90**	0.13	**0.60**	**0.22**
SS-β-TCVAE	0.51	0.19	**0.90**	0.14	**0.60**	0.20
SS-DIP-VAE.	0.25	**0.22**	**0.90**	0.15	**0.60**	0.17
SS-Factor-VAE	0.43	0.20	**0.90**	0.18	**0.60**	0.19

Table 7: Comparison of DCI, IRS, UC and CG metrics on CANDLE dataset for various models when experimented with confounded dataset where certain shapes appear in certain sizes as given in Table 8

Object	Cube	Sphere	Cylinder	Cone	Torus
Available size	Small	Small	Small	Big	Big

Table 8: Chosen confounding between object and color for experiments provided in Table 7

Model	IRS	DCI (D)	UC $\rho=1$	CG $\rho=1$	UC $\rho=2$	CG $\rho=2$
β-VAE	0.63	0.12	0.63	0.07	**0.53**	0.07
β-TCVAE	**0.75**	0.23	0.33	0.06	0.22	0.07
DIP-VAE	0.51	0.10	0.73	**0.10**	0.40	0.09
Factor-VAE	0.57	0.12	**0.86**	0.02	0.49	0.03
SS-β-VAE	0.55	0.18	0.73	0.05	0.48	0.05
SS-β-TCVAE	0.70	**0.36**	0.33	**0.10**	0.22	**0.10**
SS-DIP-VAE	0.43	0.12	0.80	0.05	0.48	0.05
SS-Factor-VAE	0.62	0.25	0.73	0.09	0.48	0.09

Table 9: Comparison of DCI, IRS, UC and CG metrics on dSprites dataset for various models when experimented with confounding dataset where where certain shapes appear in certain sizes, orientation, and position as given in Table 10

Shape	Available size	Available Orientation	vailable Position
Square	Small	$0 - \frac{2\pi}{3}$	Top Left
Ellipse	Medium	$\frac{2\pi}{3} - \frac{4\pi}{3}$	Middle
Heart	Large	$\frac{4\pi}{3} - 2\pi$	Bottom Right

Table 10: Chosen confounding between object and color for experiments provided in Table 9

Model	IRS	DCI (D)	UC $\rho=1$	CG $\rho=1$	UC $\rho=2$	CG $\rho=2$
β-VAE	0.18	**0.01**	0.28	0.11	0.19	0.11
β-TCVAE	0.38	0.008	0.00	0.10	0.00	**0.21**
DIP-VAE	0.12	0.005	0.28	0.06	0.19	0.08
Factor-VAE	0.26	**0.01**	0.66	**0.14**	**0.38**	0.13
SS-β-VAE	0.63	0.006	0.66	0.12	0.19	**0.21**
SS-β-TCVAE	**0.64**	0.007	0.28	0.06	0.19	0.12
SS-DIP-VAE	0.32	0.007	**0.76**	0.06	0.32	0.10
SS-Factor-VAE	0.50	0.006	0.00	0.12	0.00	0.20

Table 11: Comparison of DCI, IRS, UC and CG metrics on MPI3D dataset for various models when experimented with confounding dataset where confounding information is given in Table 12

Color	Shape	Size	h-axis	v-axis
Green	Cube, Cylinder	Small	0-10	0-10
Red	Cylinder, Sphere	Large	10-20	10-20
Blue	Sphere, Cube	Small, Large	20-30	20-30

Table 12: Chosen confounding between object and color for experiments provided in Table 11. All images are centered *height=1*, *background color* appear in all possible colors.

References

[1] Rui Shu, Yining Chen, Abhishek Kumar, Stefano Ermon, and Ben Poole. Weakly supervised disentanglement with guarantees. *arXiv preprint arXiv:1910.09772*, 2019. 2

[2] Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *ICML*. PMLR, 2019. 4

[3] Ran Zmigrod, Sabrina J Mielke, Hanna Wallach, and Ryan Cotterell. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. *arXiv preprint arXiv:1906.04571*, 2019. 2