10-708 Project Final: Semantically Disentangled Object Generation

Austin Dill (abdill) [1] **Neel Guha (nguha)** [1] **Rishub Jain (rishubj)** [1] **Jeffrey Li (jwl3)** [1]

Abstract

Learning disentangled latent spaces while using deep generative models remains a largely open challenge for tasks such as image generation. One narrower subset of this objective is to consider the problem of *semantic object disentanglement*, where the goal is to allow for the hierarchical generation of complex images by being able to first generate individual *objects* and then being able to compose them together. In contrast to past approaches in this domain, we develop a novel and flexible method that directly enforces such disentanglement. In practice, we observe that our method achieves strong performance on three synthetic datasets, providing very good results on quality tests for disentanglement, reconstruction, and interpretable image generation through substitution. Our results, while on relatively simplified domains, provide a strong proof of concept for our general approach.

1. Introduction

Deep generative models have proven to be incredibly adept at synthesizing high quality instances of images (Karras et al., 2017), audio recordings (Oord et al., 2016), and point clouds (Li et al., 2018) through learning a mapping of these high dimensional examples into a lower dimensional latent space. However, while these learned latent spaces provide simple sampling processes and compressed representations for complex distributions, one common hindrance to their broader utility is a lack of interpretability. That is, the relationship between an example's latent features and its original representation remains convoluted and unintelligible from a human perspective.

One commonly desired notion of interpretability for learned representations is that of *semantic disentanglement*. In this framework (Bengio et al., 2012), a learned latent vector Z would ideally decompose into independently meaningful sub-vectors $Z_1, Z_2, \ldots, Z_n$ where the substitution of any component Z_i with Z_i' would consistently alter only a single salient aspect of the decoded output (i.e. the image background, a person's eye color, etc). Having such a disentangled representation conveys many potential benefits. Not only would it make the generative model and procedure more controllable, it could also help in sidestepping the infamous mode-collapse problem (Arjovsky & Bottou, 2017) or facilitate later downstream tasks.

In this paper, we focus on the domain of images and narrow the goal of semantic disentanglement to what we call *semantic object disentanglement*. This problem is centered around the acknowledgment that many generative tasks require the composition of multiple generated objects into a meaningful whole. For example, the majority of realistic images contain more than a single person, building, or additional entity. These distinct yet major sub-parts of the overall image are what we refer to as *objects*, and their presence and interactions in an image can be quite complex. Because of this, an ideally disentangled representation would allow for image generation to be done in a hierarchical fashion: being able to (1) generate individual objects and to (2) compose them together in a realistic way.

To achieve this form of disentanglement, previous approaches have generally involved extra supervision in the training process. In such methods, the training set needs to be augmented with many variations of labelled object compositions and extra penalties and used to enforce disentanglement (Donahue et al., 2017). However, not only do these approaches suffer from inconsistent performance, finding reconstruction and disentanglement hard to jointly perform, they also are expensive from a data collection and labelling point of view. In contrast to these approaches, we propose a new method which avoids these issues by enforcing semantic object disentanglement in a strict fashion, while also removing the need for an extensive paired dataset.

Specifically, we build off an autoencoder framework, where the key assumption is that the latent representation for a complex image can be subdivided into features that correspond to (1) the identities of individual objects and (2) the contextual attributes (eg. spatial and rotational information) for these objects. During our training process, given

[1] Carnegie Mellon University, Pittsburgh, PA 15213, USA.

10708 Class Project, Spring, 2019.

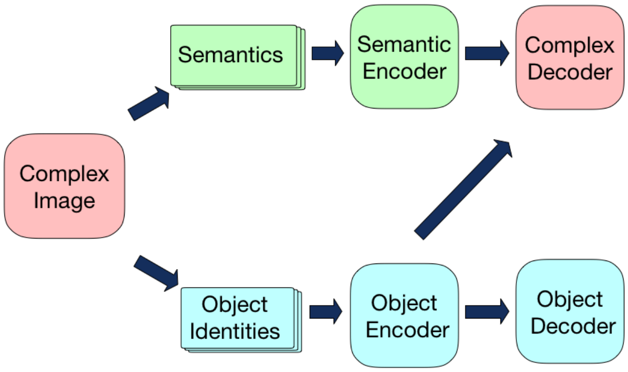

Figure 1. Our ultimate goal is to create a low dimensional, disentangled representation of the input , separated by its constituent objects and its semantic information.

knowledge about the existence of an image's objects, we can enforce the desired feature separation by encoding information about the individual objects separately. That is, we distinctly encode identity information (using raw sub-images) and semantic information (using representations such as spatial masks) for each object, before concatenating all sub-vectors into a final representation. Then, we ensure that this representation is useful by allowing our decoder to only access to information from this separately generated final vector. If successful, we can then potentially generate images that control and successfully compose the exact objects we want to include. Also, though we work with autoencoders, our approach can be applied to any underlying generative model.

In the following sections, we first introduce the existing work in this field. Second, we articulate in greater detail our proposed approach, labelling the multiple high-level components and describing their details more carefully. Finally we present some empirical evaluations of our proposed method on three synthetic datasets of composite images, each generated using base images from MNIST, Fashion MNIST, and Fruit 360. Although these datasets involve some key simplifications from fully natural images, we observe that the latent spaces learned by our method achieves promising results compared to natural baselines for disentanglement, reconstruction, and interpretable novel image generation.

2. Related Works

Deep Autoencoders (Baldi, 2011) and Generative Adversarial Networks (GANS) (Goodfellow et al., 2014) have been successful in generating realistic synthetic data samples in a wide range of settings (Lin et al., 2014; Oord et al., 2016; Karras et al., 2017). However, the latent spaces ingested by these models are rarely interpertable or well understood without explicit class based conditioning (Mirza & Osindero (2014). In the ensuing section, we review existing approaches for disentangled representations and semantic decomposition of latent spaces.

Semantic decomposition of representations shares many similarities to *style transfer* (Gatys et al., 2015) or *novel view synthesis* (Avidan & Shashua, 1997). In general, prior work attempts to explicitly disentangle a representation into dual components (i.e. style and content, time and content, object-1 and object-2, and etc), or transfer one component (i.e. style) from one input to another.

Zhu et al. (2017) propose Cycle-Consistent Adversarial Networks which allow for style transfer by incorporating additional structure and regularization by combining the normal *adversarial losses* with *cycle consistency losses* which ensure $F(G(X)) \approx X$ and $G(F(Y)) \approx Y$ (where $G : X \rightarrow Y$ and $F : X \rightarrow Y$).

Analgously, Azadi et al. (2018) developed Compositional-GAN as a method to perform object composition in conditional image generation, generating images from two latent sources *without any prior information about the scene layout*. For two objects, given sets of images from each's marginal distribution, X and Y, as well as a set from their joint distribution (containing both objects), C, this method is able to generate realistic composite images containing both objects. The variations of their method were able to handle both paired training data (elements in C directly correspond to a pair of elements from X and Y) and unpaired.

From a more representation centered perspective, Denton & Birodkar (2017) present DrNET, a model for learning disentangled representations from video through a predictive autoencoder that factors each video frame's latent representation into two parts: 1) **content**: a *time independence* component that remains constant through the entire clip, and 2) **pose**: a *time dependent* component that captures dynamic aspects of the clip. Donahue et al. (2017) bears the closest resemblance to our work, and analyze semantic decomposition in the context of facial photograph generation. They propose Semantically Decomposed GANs (SD-GANs), in which the latent space $\mathcal{Z}$ is decomposed into subspaces $\mathcal{Z}_I$ and $\mathcal{Z}_O$, corresponding to the *identity* (i.e. the person in the photograph) and *observational* (i.e. lighting, pose, etc) portions of an image respectively. SD-GANs utilize a pairwise training scheme with siamese networks (Bromley et al., 1993; Chopra et al., 2005), in which each sample from the dataset corresponds to two images with common Z_I and distinct Z_O. The discriminator is trained to reject pairs if 1) the generated images are not photorealistic or 2) the generated images correspond to different identities. Our work differs in several ways. First, our segmentation-encoding-decoding architecture allows for decomposition over multiple objects (not just observation vs identity). Secondly, rather than relying on siamese networks to enforce consistency in the latent space, we use a shared encoder (described further below).

Disentanglement can also be achieved by reweighting the traditional encoding/decoding objective. (Chen et al., 2016) present InfoGan, which learns disentangled representations by maximizing the mutual information between a small subset of the latent variables and the observation. (Higgins et al., 2017) propose $\beta-$VAE, which encourages independence between latent variables. (Chen et al., 2018) show that this success can be explained via a decomposition of the variational lower bound. (Kim & Mnih, 2018) improve on the reconstruction quality of $\beta-$VAE by augmenting the objective with a penalty that encourages the marginal distribution of representations to be factorial.

3. Methodology

Broadly, we propose a scheme for disentangling the latent space by independently learning latent representations for the objects in an image and their spatial relationships (semantics). A high-level overview of our method is illustrated by 1.

Formally, our goal is to learn a disentangled latent space $\mathcal{Z}$ (over $\mathbb{R}^s$), a corresponding encoding function $E : \mathcal{X} \to \mathcal{Z}$, and generative function $G : \mathcal{Z} \to \mathcal{X}$. In this work, we consider $x \in \mathcal{X}$ (over $\mathbb{R}^d$ to be an image consisting of multiple distinct *objects*. The exact definition of an object depends on the task setting. In a natural scene of a dining table, objects could correspond to distinct items (silverware, individuals, etc). In a human profile picture, objects could correspond to elements of the photo (hair, ears, eyes, etc).

In order for $Z \in \mathcal{Z}$ to be 'disentangled', we should be able to decompose Z into non-overlapping subspaces $Z^1, ..., Z^k \subseteq Z$ such that each Z^i corresponds to a distinct aspect of the generated image $G(Z)$. For clarity, we refer to each subspace as a *group* G_i, consisting of the indices over which the subspace exists. Given two latent vectors z_1 and z_2 differing only by group G_i, we should expect the generated images $G(z_1)$ and $G(z_2)$ to only differ by the aspect controlled by G_i. In this work, we disentangle representations based on 1) the objects in the image, and 2) the spatial information contained in the images. Hence, for a $k-$object image, $\mathcal{Z}$ is divided into $k+1$ groups, corresponding to the k objects ($G_1, ..., G_k$ and G_{k+1} corresponding to spatial information.

We learn disentangled representations by solving a reconstruction task over multi-object images. Rather than learning this reconstruction in an end-to-end fashion, we learn reconstructions over the distinct groups $G_1, \cdot, G_{k+1}$. We now describe this process:

1. Object Detection: Given a multi-object image x_i, we apply an image segmentation technique such as Mask R-CNN (He et al., 2017) to x_i in order to identify the k objects contained in the image. Let x_i^j denote the image corresponding to the bounding box around object j. We rescale $x_i^1, ..., x_i^k$ to a fixed size.

In this work, we assume that image objects have already been identified and that $x_i^1, ..., x_i^k$ is provided. Though this simplifies our task, the challenge of identifying $x_i^1, .., x_i^k$ is primarily a segmentation/object identification problem, and beyond the scope of this project.

2. Latent Object Representation: We derive independent latent representations for each identified object by solving an object-specific reconstruction task (corresponding to the light blue boxes in Figure 1). Broadly, our goal is to learn a latent representation from which we reconstruct the object. This representation should be completely independent of any spatial information contained in the original image. Applying the encoder to each object $x_i^1, \cdots, x_i^k$ yields a set of latent object specific representations $z_i^1, \cdots, z_i^k$.

3. Spatial Information: We explore simple methods for capturing spatial relationships between objects in the original image (e.g. the *semantics*). In this work, we consider two types of spatial information: relative positions and orientation.

In order to capture the relative positions of different objects, we leverage the original bounding boxes identified in Step 1 from our image segments/object detector. We construct an image *mask*, where pixels within the same bounding boxes are assigned a fixed value, and all pixels not contained in any bounding box are left blank. Figure 2 provides an illustrated example of this mask. In practice, we find that is important to assign each bounding box to a different pixel value (shading color). For simplicity, we use pixel values of 1.0, 0.75 and 0.5 for the three different bounding boxes in our setup.

We can use a similar approach for capturing the orientation of objects in the original image (e.g. their rotational angle). For simplicity, we assume that the orientation angle of the object is known[1]. We construct an angle mask by first constructing a blank image with a colored arrow stretching from the bottom edge to the top edge. We then rotate this mask by the rotation angle for the object in the original image, so that the head of the arrow in the mask aligns with the top of the rotated object. Figure 5 provides an example of this rotational mask.

In either case, we can flatten the generated mask and denote it by the vector z_{k+1}. In practice, we've found that rep-

[1] Given a base image angle, we can rotate the base image until it most closely resembles the object's orientation in the image, thus giving us an estimate of the angle.

resenting spatial information pictorally (i.e. with a mask) produces far better results than simply encoding the (x,y) positions of the bounding boxes. This is an interesting question for future work.

More generally, we note that z_{k+1} could correspond to any latent representation of spatial information contained in the multi-object image, and could conceivably be generated by a spatially focused encoder. We hope to explore this more in future work.

4. Embedding Combination: Given the single object latent representations $z_i^1, \cdots, z_i^k$ and the spatial information representation $z_{k=1}$, we construct a combined multi-object latent representation $Z_i = \left[z_i^1, ..., z_i^{k+1}\right]$. The ordering of object vectors z_i^j must align with the pixel values. Hence, the z_i^1 must correspond to the object in the location specified by the bounding box with pixel values of 1.0, z_i^2 must correspond to the object in the location specified by the bounding box with pixel values of 0.75, and etc.

5. Reconstruction: Finally, we learn a decoder G by minimizing the reconstruction loss when computing $G(Z)$. Importantly, the loss from $G(Z)$ is not propagated to the object specific encoder. In essence, this ensures the disentanglement in our latent space. If the multi-object reconstruction loss were propagated to the object-specific encoder, the object encoders may inadvertently learn spatial relationships between the objects, thereby increasing 'entanglement'.

4. Experimental Methodology

4.1. Problem Settings

In order to determine the feasibility of our approach, we tested a set of varying complexity disentanglement problems by altering the complexity of the underlying objects. In section 4.4 we examine our model's ability to disentangle spatial information. In section 4.5 we examine our model's ability to disentangle orientation-based semantic information.

4.2. Evaluation Metrics

We will explore our model through the lens of reconstruction quality, level of disentanglement, and a substitution test.

Reconstruction quality will be measured both in terms of reconstruction loss when compared to a non-disentangled model and in terms of a classifier test where a pre-trained object detector is run on the reconstructed images. A high accuracy on this test set will indicate that our reconstruction has high fidelity, as the generated image qualitatively contains the right kind of object in the right location.

Input	$H \times W \times C$ image
Conv Layer	64, (5,5), (2,2)
Activation	Leaky ReLU
Conv Layer	128, (5,5), (2,2)
Activation	Leaky ReLU
Dense Layer	-
Output	1×100 latent code

Table 1. Object Encoder Architecture

Disentanglement will be measured through a classification task. We will train a support vector machine on image encodings and try to predict both the object class and the discretized semantic "class" on both the object encodings and the semantic encodings. We will consider our model to have successfully disentangled these concepts if a trained SVM is able to get high accuracy using object (semantic) encodings for object (semantic) classification and near-random accuracy when using semantic (object) encodings for object (semantic) classification.

Finally, we will test the model's ability for "substitution". We define substitution to be a latent encodings ability to have a subsection swapped with a different encoding and to produce the desired outcome. This will be measured in the same way that reconstruction quality was measured.

These three qualities should imply that not only is our generative model able to generalize, but that it has successfully disentangled the desired concepts.

4.3. Model Details

Our design is to pair a convolutional object encoder and a convolution semantic encoder with two separate deconvolutional decoders, one for simple object generation and another for multi-object/semantically complex generation. We reason that the encoder will be forced to limit object information to a single sub-vector by the simple object decoder which will in turn force the complex decoder to respect the separation of object encodings.

For reproducibility, we have included the individual model architectures in **Tables 2, 3** and **4**. Each convolutional layer has its number of channels, kernel size, and stride size presented in the second column. Let H, W, C represent a single object image's height, width, and number of channels respectively. Let H', W', C' represent a complex image's height, width, and number of channels. Finally, let n represent the number of objects in the complex image. All non-included parameters are tuned to match the requisite output/input shapes.

The loss function used was a standard Mean Square Error (MSE) used to train autoencoders, with the loss applied to the discrepancy between the input and the recon-

Input	$H' \times W' \times C'$ image
Conv Layer	64, (5,5), (2,2)
Activation	Leaky ReLU
Conv Layer	128, (5,5), (2,2)
Activation	Leaky ReLU
Dense Layer	-
Output	1×100 latent code

Table 2. Semantic Encoder Architecture

Input	1×100 latent code
Dense Layer	$1 \times 12,544$ output
Batch Norm	-
Activation	Leaky ReLU
DeConv Layer	128, (5,5), $(\cdot, \cdot)$
Batch Norm	-
Activation	Leaky ReLU
DeConv Layer	64, (5,5), $(\cdot, \cdot)$
Batch Norm	-
Activation	Leaky ReLU
DeConv Layer	1, (5,5), $(\cdot, \cdot)$
Activation	ReLU
Output	$H \times W \times C$ image

Table 3. Single Object Decoder Architecture

Input	$(n + 1) \times 100$ latent code
Dense Layer	$1 \times (\cdot)$ output
Batch Norm	-
Activation	Leaky ReLU
DeConv Layer	128, (5,5), $(\cdot, \cdot)$
Batch Norm	-
Activation	Leaky ReLU
DeConv Layer	64, (5,5), $(\cdot, \cdot)$
Batch Norm	-
Activation	Leaky ReLU
DeConv Layer	1, (5,5), $(\cdot, \cdot)$
Activation	ReLU
Output	$H' \times W' \times C$ image

Table 4. Complex Decoder Architecture

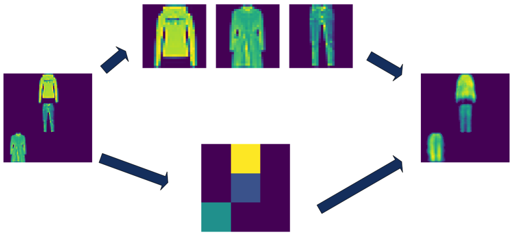

Figure 2. In order to encode a complex spatial image, we split it into object images and a spatial mask. These components are used to generate the reconstructed image on the right.

structed images. We used the Adam optimizer (Kingma & Ba, 2015) and the gradient was allowed to flow from both decoders to the shared encoder. All of this was implemented in the Tensorflow framework (Abadi et al., 2016) and every experiment was run on a GTX 1060 GPU with 6GB of memory. The training for each of the results below were achieved with 10,000 iterations with a batch size of 256 for both input datasets.

4.4. Spatial Setting

In order to examine our models ability to disentangle an *object* encoding from a *spatial* encoding, we created two datasets, one made of MNIST digits and the other made of Fashion MNIST articles of clothing. We added spatial variability by random selecting squares on a 3×3 grid to be filled with randomly selected objects from these datasets. We then provided the object encoder with the cropped object images and the semantic encoder with the spatial mask. The pipeline for reconstructing such an image can be seen in Figure 2.

4.4.1. Reconstruction

As can be seen in Table 5, our reconstruction error is considerably higher on unseen data than the typical non-disentangled autoencoder. While on its own this would be troubling, our classification results in Table 6 and the randomly selected instances in Figure 3 show that this may be caused by the fragility of MSE to small perturbations, such as due to object translation, and not a true weakness of our model.

Spatial Reconstruction	MSE
MNIST-Baseline	0.006449355
MNIST-Spatial	2405.2346
fMNIST-Baseline	0.0069138384
fMNIST-Spatial	4510.6846

Table 5. Reconstruction error on held-out test set.

Spatial Reconstruction	Classification accuracy
MNIST-Baseline	0.862
MNIST-Spatial	0.910
fMNIST-Baseline	0.546
fMNIST-Spatial	0.586

Table 6. Classification accuracy on held-out test set.

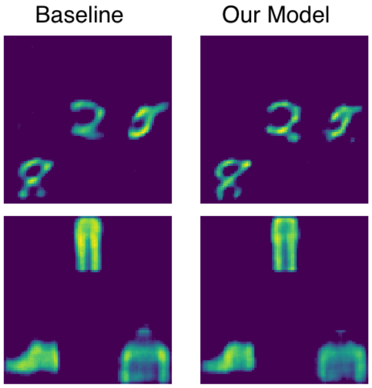

Figure 3. Though the reconstruction loss is markedly different between the baseline and our model, the qualititive results show little difference.

4.4.2. DISENTANGLEMENT

In order for a encoding to be consider disentangled, it must contain little if any information about the features it is ”disentangled” from. One way to test this is to train a classifier on encodings for both the feature they are supposed to predict and the features they shouldn’t be able to predict. For these tests, we trained a simple SVM to predict either the object class (i.e. which digit for MNIST and which article of clothing for Fashion-MNIST) and a discretized version of their spatial information. To discretize the spatial information, we selected 10 spatial masks at random and generated a dataset including only these masks with each label corresponding to the image’s spatial mask.

ID Encodings	Train	Test
MNIST-BASELINE	0.9952	0.978
MNIST-Spatial	0.9975	0.9768
fMNIST-BASELINE	0.9480	0.8620
fMNIST-Spatial	0.9739	0.8628

Table 7. Classification accuracy for object IDs using object embeddings.

It can be seen in Table 7 that our object identity encodings match the performance of a non-disentangled version in both training and testing. This indicates that our model contains as much information about the object class as the baseline and performs significantly better than randomly guessing a class.

Spatial Encodings	Train	Test
MNIST-BASELINE	0.9587	0.8960
MNIST-Spatial	1	1
fMNIST-BASELINE	1	1
fMNIST-Spatial	1	1

Table 8. Classification accuracy for spatial class using semantic embeddings.

Table 8 shows that our spatial embeddings contain enough information to perform as well as the baseline in this simplified case. Of note is that the MNIST-Spatial encoding outperforms its baseline. This could be because the baseline encodings are forced to devote more attention to the identity encoding and does not retain as much spatial information as a dedicated encoding.

ID Encodings	Train	Test
MNIST-BASELINE	0.9587	0.8960
MNIST	1	0.12
fMNIST-BASELINE	1	1
fMNIST	1	0.112

Table 9. Classification accuracy for spatial class using object embeddings.

Finally, our object ID encodings contain nearly no information about the spatial information, as can be seen by the test accuracy in Table 9. This stands in stark contrast to the completely accurate predictions from the spatial encodings discussed about.

These tests taken together bolster the intuition that our architecture disallows any mixing of object identity information and spatial semantic information.

4.4.3. SUBSTITUTION

Figure 10

Spatial Reconstruction after Substitution	Classification accuracy
Ground Truth	0.862
MNIST-Spatial	0.910
fMNIST-Baseline	0.546
fMNIST-Spatial	0.586

Table 10. Classification accuracy on held-out test set after substituting an object identity vector.

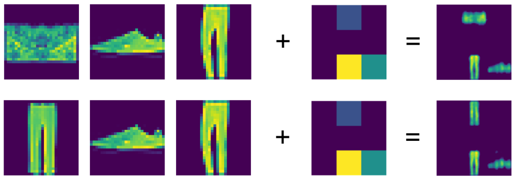

Figure 4. Our model's disentangled latent space allows for the substitution of one encoding for another without altering the rest of the generated image.

4.5. Orientation Setting

In order to test our model's ability to disentangle orientation information from object identity information, we created a dataset consisting of rotated MNIST digits. Though we provide the angular information since we generated the rotated images, this could be more generally done using a technique such as RotNets (Gidaris et al., 2018). We then provided the object encoder with the reoriented object image and the semantic encoder with the angle mask. The pipeline for reconstructing such an image can be seen in Figure 5.

4.5.1. Reconstruction

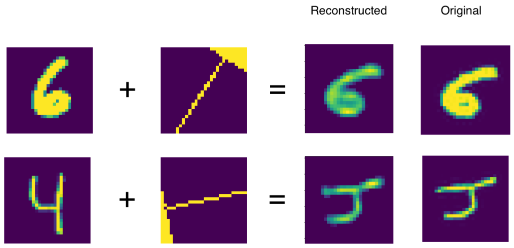

Figure 5. In order to encode an oriented image, we split it into an object image and an angle mask. These components are used to generate the reconstructed image on the right.

Though the disparity is less drastic than in the previous setting, our reconstruction loss is still considerably higher than the baseline in Table 11. This gives us reason to once again inspect randomly selected images in Figure 6. We did not train a classifier for this reconstruction task due to the relatively minor difference in reconstruction error.

Angle Reconstruction	**MSE**
MNIST-Baseline	0.0036420375
MNIST-Angle	0.014301385

Table 11. Reconstruction error on held-out test set.

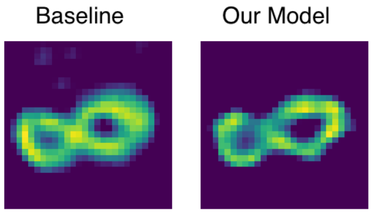

Figure 6. While the reconstruction error is worse with our model, the reconstructed images preserve finer detail.

4.5.2. Disentanglement

Much like our spatial disentanglement tests, our orientation tests show that while the encodings of relevant features are highly predictive of their encoded feature while non-predictive of other features.

ID Encodings	**Train**	**Test**
MNIST-BASELINE	0.9952	0.9780
MNIST-Rotation	0.9985	0.9736

Table 12. Classification accuracy for object IDs using object embeddings.

Angle Encodings	**Train**	**Test**
MNIST-BASELINE	0.9784	0.9212
MNIST-Rotation	1	1

Table 13. Classification accuracy for angle class using angle embeddings.

ID Encodings	**Train**	**Test**
MNIST-BASELINE	0.9784	0.9212
MNIST-Rotation	0.6649333333	0.1184

Table 14. Classification accuracy for angle class using object embeddings.

4.5.3. Substitution

Without the trainer classifier for our substitution test, we are left with the qualitative task of judging our reconstructions when sections of the latent code have been substituted for each other. An example of our results can be seen in Figure 7.

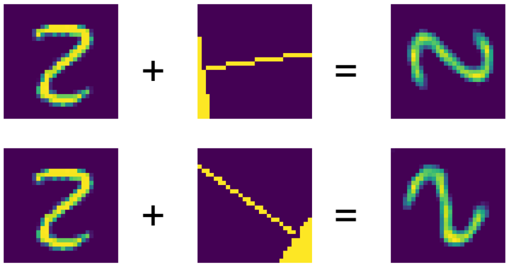

Figure 7. Our model allows for the plausible generation of a novel combination of a object/orientation pair unseen in training.

4.6. Color Images

While our experimentation with color images are too preliminary to be detailed as thoroughly as the experiments above, our early results are promising as can be seen in Figure 8. While these results can be achieved simply by changing the channels on each network to accommodate color, we hope to expand our techniques detailed in this paper to separate color information from object identity.

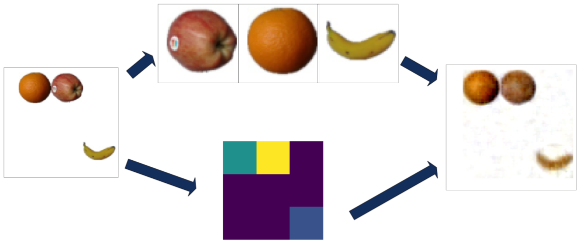

Figure 8.

5. Conclusions

Our initial results show promise for the use of deep generative models to learn a latent space encoding that allows for the joint generation of individual and multi-object instances. Extracting features in a hierarchieal fashion allows us to learned semantic encodings without the need for large paired datasets to distinguish between features of interest. Furthermore, by forcing our model to learn a completely disentangled latent representation, we are able to perform the highly desirable task of substitution without a loss in generative quality.

While in this paper we investigated only simplified versions of our ultimate goal, we have shown that disentangling semantic objects is possible when spatial or angular information is given. In our next steps, we hope to extend these techniques to more complex datasets, featuring a variable number of objects and more intricate composition. In addition, our ultimate hope is for this to develop into a general framework for semantic object disentanglement, applicable regardless of generative model or domain.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.

Arjovsky, M. and Bottou, L. Towards Principled Methods for Training Generative Adversarial Networks. *arXiv e-prints*, art. arXiv:1701.04862, Jan 2017.

Avidan, S. and Shashua, A. Novel view synthesis in tensor space. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1034–1040. IEEE, 1997.

Azadi, S., Pathak, D., Ebrahimi, S., and Darrell, T. Compositional gan: Learning conditional image composition. 2018.

Baldi, P. Autoencoders, unsupervised learning and deep architectures. In *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning Workshop - Volume 27*, UTLW'11, pp. 37–50. JMLR.org, 2011. URL http://dl.acm.org/citation.cfm?id=3045796.3045801.

Bengio, Y., Courville, A. C., and Vincent, P. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR*, abs/1206.5538, 2012. URL http://arxiv.org/abs/1206.5538.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. Signature verification using a ”siamese” time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, pp. 737–744, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=2987189.2987282.

Chen, T. Q., Li, X., Grosse, R. B., and Duvenaud, D. K. Isolating sources of disentanglement in variational autoencoders. *CoRR*, abs/1802.04942, 2018. URL http://arxiv.org/abs/1802.04942.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *CoRR*, abs/1606.03657, 2016. URL http://arxiv.org/abs/1606.03657.

Chopra, S., Hadsell, R., LeCun, Y., et al. Learning a similarity metric discriminatively, with application to face verification. In *CVPR (1)*, pp. 539–546, 2005.

Denton, E. and Birodkar, V. Unsupervised learning of disentangled representations from video. *CoRR*, abs/1705.10915, 2017. URL http://arxiv.org/abs/1705.10915.

Donahue, C., Balsubramani, A., McAuley, J., and Lipton, Z. C. Semantically decomposing the latent spaces of generative adversarial networks. *CoRR*, abs/1705.07904, 2017. URL http://arxiv.org/abs/1705.07904.

Gatys, L. A., Ecker, A. S., and Bethge, M. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015. URL http://arxiv.org/abs/1508.06576.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. *CoRR*, abs/1803.07728, 2018. URL http://arxiv.org/abs/1803.07728.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. 2014.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Kim, H. and Mnih, A. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

Kingma, D. and Ba, J. Adam: a method for stochastic optimization (2014). *arXiv preprint arXiv:1412.6980*, 15, 2015.

Li, C.-L., Zaheer, M., Zhang, Y., Poczos, B., and Salakhutdinov, R. Point cloud gan. *arXiv preprint arXiv:1810.05795*, 2018.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Mirza, M. and Osindero, S. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. URL http://arxiv.org/abs/1411.1784.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. 2017.