Guiding InfoGAN with Semi-supervision

Adrian Spurr(✉), Emre Aksan, and Otmar Hilliges

Advanced Interactive Technologies, ETH Zurich, Zurich, Switzerland
{adrian.spurr,emre.aksan,otmar.hilliges}@inf.ethz.ch

Abstract. In this paper we propose a new semi-supervised GAN architecture (ss-InfoGAN) for image synthesis that leverages information from *few* labels (as little as 0.22%, max. 10% of the dataset) to learn semantically meaningful and controllable data representations where latent variables correspond to label categories. The architecture builds on Information Maximizing Generative Adversarial Networks (InfoGAN) and is shown to learn both continuous and categorical codes and achieves higher quality of synthetic samples compared to fully unsupervised settings. Furthermore, we show that using *small* amounts of labeled data speeds-up training convergence. The architecture maintains the ability to disentangle latent variables for which no labels are available. Finally, we contribute an information-theoretic reasoning on how introducing semi-supervision increases mutual information between synthetic and real data. Code related to this chapter is available at: https://github.com/spurra/ss-infogan.

1 Introduction

In many machine learning tasks it is assumed that the data originates from a generative process involving complex interaction of multiple independent factors, each accounting for a source of variability in the data. Generative models are then motivated by the intuition that in order to create realistic data a model must have "understood" these underlying factors. For example, images of handwritten characters are defined by many properties such as character type, orientation, width, curvature and so forth.

Recent models that attempt to extract these factors are either completely supervised [18,20,23] or entirely unsupervised [3,5]. Supervised approaches allow for extraction of the desired parameters but require fully labeled datasets and a priori knowledge about which factors underlie the data. However, factors not corresponding to labels will not be discovered. In contrast, unsupervised approaches require neither labels nor a priori knowledge about the underlying factors but this flexibility comes at a cost: such models provide no means of exerting control on what kind of features are found. For example, Information Maximizing Generative Adversarial Networks (InfoGAN) have recently been shown to

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-319-71249-9_8) contains supplementary material, which is available to authorized users.

M. Ceci et al. (Eds.): ECML PKDD 2017, Part I, LNAI 10534, pp. 119–134, 2017.
https://doi.org/10.1007/978-3-319-71249-9_8

learn disentangled data representations. Yet the extracted representations are not always directly interpretable by humans and lack direct measures of control due to the unsupervised training scheme. Many application scenarios however require control over *specific* features.

Embracing this challenge, we present a new semi-supervised generative architecture that requires only few labels to provide control over which factors are identified. Our approach can exploit already existing labels or use datasets that are augmented with easily collectible labels (but are not fully labeled). The model, based on the related InfoGAN [3] is dubbed semi-supervised InfoGAN (ss-InfoGAN). In our approach we maximize two mutual information terms: (i) The mutual information between a code vector and real labeled samples, guiding the corresponding codes to represent the information contained in the labeling, (ii) and the mutual information between the code vector and the synthetic samples. By doing so ss-InfoGAN can find representations that unsupervised methods such as InfoGAN fail to find, for example the category of digits of the SVHN dataset. Notably our approach requires only 10% of labeled data for the hardest dataset we tested and for simpler datasets only 132 labeled samples (0.22%) were necessary.

We discuss our method in full, provide an information theoretical rationale for the chosen architecture and demonstrate its utility in a number of experiments on the MNIST [13], SVHN [19], CelebA [14] and CIFAR-10 [10] datasets. We show that our method improves results over the state-of-the-art, combining advantages of supervised and unsupervised approaches.

2 Related Work

Many approaches to modeling the data generating process and identifying the underlying factors by learning to synthesize samples from disentangled representations exist. An example of an early approach is supervised bi-linear models [27], separating style from the content. Zhu et al. [29] use a multi-view deep perceptron model to untangle the identity and viewpoint of face images. Weakly supervised methods based on supervised clustering, have been proposed such as high-order Boltzman machines [22] applied on face images.

Variational Autoencoders (VAEs) [9] and Generative Adversarial Networks (GANs) [6] have recently seen a lot of interest in generative modeling problems. In both approaches a deep neural network is trained as a generative model by using standard backpropagation, enabling synthesis of novel samples without explicitly learning the underlying data distribution. VAEs maximize a lower bound on the marginal likelihood which is expected to be tight for accurate modeling [2,8,24,26]. In contrast, GANs optimize a minimax game objective via a discriminative adversary. However, they have been shown to be unstable and fragile [17,23].

Employing semi-supervised learning, Kingma et al. [7] use VAEs to isolate content from other variations, and achieve competitive recognition performance in addition to high-quality synthetic samples. Deep Convolutional Inverse Graphics Network (DC-IGN) [11], which uses a VAE architecture and a specially tailored training scheme is capable of learning a disentangled latent space in fully supervised manner. Since the model is evaluated by using images of 3D models, labels for the underlying factors are cheap to attain. However, this type of dense supervision is unfeasible for most non-synthetic datasets.

Adversarial Autoencoders [15] combine the VAE and GAN frameworks in using an adversarial loss on the latent space. Similarly, Mathieu et al. [16] introduces an adversarial loss on the reconstructions of VAE, that is, on the pixel space. Both models are shown to learn both discrete and continuous latent representations and to disentangle style and content in images. However, these hybrid architectures have conceptually different designs as opposed to GANs. While the former learns the data distribution via Autoencoder training and employ the adversarial loss as a regularizer, the latter directly relies on an adversarial objective. Despite the robust and stable training, VAEs have tendency to generate blurry images [12].

Conditional GANs [18,20,23] augment the GAN framework by using class labels. Mirza and Osindero [18] train a class-conditional discriminator while [20,23] use auxiliary loss terms for the labels. Salimans et al. [23] use conditional GANs for pre-training, aiming to improve semi-supervised classification accuracy of the discriminator. Similarly, the AC-GAN model [20] introduces an additional classification task in the discriminator to provide class-conditional training and inference of the generator in order to be able to synthesize higher resolution images than previous architectures. Our work is similar to the above in that it provides class-conditional generation of images. However, due to MI loss terms our architecture can (i) be employed in both supervised *and* semi-supervised settings, (ii) can learn interpretable representations in addition to smooth manifolds and (iii) can exploit continuous supervision signals if such labels are available.

Comparatively fewer works treat the subject of fully unsupervised generative models to retrieve interpretable latent representations. Desjardins et al. [5] introduced a higher-order RBM for recognition of facial expressions. However, it can only disentangle discrete latent factors and the computational complexity rises exponentially in the number of features. More recently, Chen et al. [3] developed an extension to GANs, called Information Maximizing Generative Adversarial Networks (InfoGAN). It enforces the generator to learn disentangled representations through increasing the mutual information between the synthetic samples and a newly introduced latent code. Our work extends InfoGAN such that additional information can be used. Supervision can be a necessity if the model struggles in learning desirable representations or if *specific* features need to be controlled by the user. Our model provides a framework for semi-supervision in InfoGANs. We find that leveraging few labeled samples brings improvements on the convergence rate, quality of representations and synthetic samples. Moreover, semi-supervision helps the model in capturing otherwise difficult to capture representations.

3 Method

3.1 Preliminaries: GAN and InfoGAN

In the GAN framework, a generator G producing synthetic samples is pitted against a discriminator D that attempts to discriminate between real data and samples created by G. The goal of the generator is to match the distribution of generated samples P_G with the real distribution P_{data}. Instead of explicitly estimating $P_G(x)$, G learns to transform noise variables $z \sim P_{noise}$ into synthetic samples $\tilde{x} \sim P_G$. The discriminator D outputs a single scalar $D(x)$ representing the probability of a sample x coming from the true data distribution. Both $G(z; \theta_g)$ and $D(x; \theta_d)$ are differentiable functions parametrized by neural networks. We typically omit the parameters θ_g and θ_d for brevity. G and D are simultaneously trained by using the minimax game objective $V_{GAN}(D, G)$:

$$\min_G \max_D V_{GAN}(D, G) = \mathbb{E}_{x \sim P_{data}}[\log D(x)] + \mathbb{E}_{z \sim P_{noise}}[\log(1 - D(G(z)))] \quad (1)$$

GANs map from the noise space to data space without imposing any restrictions. This allows G to produce arbitrary mappings and to learn highly dependent factors that are hard to interpret. Therefore, variations of z in any dimension often yields entangled effects on the synthetic samples $\tilde{x}$. InfoGANs [3] are capable of learning disentangled representations. InfoGAN extends the unstructured noise z by introducing a latent code c. While z represents the incompressible noise, c describes semantic features of the data. In order to prevent G from ignoring the latent codes c, InfoGAN regularizes learning via an additional cost term penalizing low mutual information between c and $\tilde{x} = G(z, c)$:

$$\begin{aligned} \min_{G,Q} \max_D V_{InfoGAN}(D, G, Q, \lambda_1) &= V_{GAN}(D, G) - \lambda_1 L_I(G, Q), \\ I(C; \tilde{X}) \geq L_I(G, Q) &= \mathbb{E}_{c \sim P_c, \tilde{x} \sim P_G}[\log Q(c|\tilde{x})] + H(c), \end{aligned} \quad (2)$$

where Q is an auxiliary parametric distribution approximating the posterior $P(c|x)$, L_I corresponds to the lower bound of the mutual information $I(C; \tilde{X})$ and λ_1 is the weighting coefficient.

3.2 Semi-supervised InfoGAN

Although InfoGAN can learn to disentangle representations in an unsupervised manner for simple datasets, it struggles to do so on more complicated datasets such as CelebA or CIFAR-10. In particular, capturing categorical codes is challenging and hence InfoGAN yields poorer performance in class-conditional generation task than competing methods. Moreover, depending on the initialization, the learned latent codes may differ between training sessions further reducing interpretability.

In Semi-Supervised InfoGAN (ss-InfoGAN) we introduce available or easily acquired labels to address these issues. Figure 1 schematically illustrates our architecture. To make use of label information we decompose the latent code

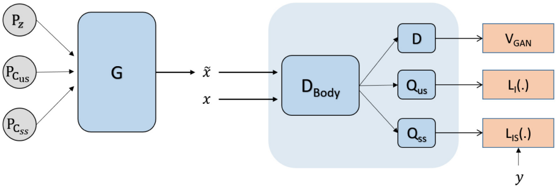


Fig. 1. Schematic overview of the ss-InfoGAN network architecture. P_z, $P_{C_{us}}$ and $P_{C_{ss}}$ are the distributions of the noise and latent variables z, c_{us} and c_{ss}, respectively.

c into a set of semi-supervised codes, c_{ss}, and unsupervised codes, c_{us}, where $c_{ss} \cup c_{us} = c$. The semi-supervised codes encode the same information as the labels y, whereas c_{us} are free to encode potential remaining semantic factors.

We seek to increase the mutual information $I(C_{ss}; X)$ between the latent codes c_{ss} and the labeled real samples x, by interpreting labels y as the latent codes c_{ss}, (i.e. $y = c_{ss}$). Note that not all samples need to be labeled for the generator to learn the inherent semantic meaning of y. We additionally want to increase the mutual information $I(C_{ss}; \tilde{X})$ between the semi-supervised latent codes and the synthetic samples $\tilde{x}$ so that information can flow back to the generator. This is accomplished via Variational Information Maximization [1] in deriving lower bounds for both MI terms. For the lower bounds of $I(C_{ss}; \cdot)$ we utilize the same derivation as InfoGAN:

$$I(C_{ss}; X) \geq \mathbb{E}_{c\sim P_{C_{ss}}, x\sim P_X}[\log Q_1(c_{ss}|x)] + H(C_{ss}) = L^1_{IS}(Q_1), \tag{3}$$

$$I(C_{ss}; \tilde{X}) \geq \mathbb{E}_{c\sim P_{C_{ss}}, \tilde{x}\sim P_G}[\log Q_2(c_{ss}|\tilde{x})] + H(C_{ss}) = L^2_{IS}(Q_2, G), \tag{4}$$

where Q_1 and Q_2 are again auxiliary distributions to approximate posteriors and are parametrized by neural networks. With $Q_1 = Q_2 = Q_{ss}$ we attain the MI cost term:

$$L_{IS}(Q_{ss}, G) = L^1_{IS}(Q_{ss}) + L^2_{IS}(Q_{ss}, G) \tag{5}$$

Since we would like to encode the labels y via latent codes c_{ss}, we optimize $L^1_{IS}(Q_{ss})$ with respect to Q_{ss} and $L^2_{IS}(Q_{ss}, G)$ only with respect to G. The final objective function is then:

$$\min_{G, Q_{us}, Q_{ss}} \max_{D} \quad V_{ss\text{-}InfoGAN}(D, G, Q_{us}, Q_{ss}, \lambda_1, \lambda_2) \tag{6}$$

$$= V_{InfoGAN}(D, G, Q_{us}, \lambda_1) - \lambda_2 L_{IS}(G, Q_{ss}) \tag{7}$$

Training Q_{ss} on labeled real data (x, y) enables Q_{ss} to encode the semantic meaning of y via c_{ss} by means of increasing the mutual information $I(C_{ss}; X)$. Simultaneously, the generator G acquires the information of y indirectly by increasing $I(C_{ss}; \tilde{X})$ and learns to utilize the semi-supervised representations in synthetic samples. In our experiments we find that a small subset of labeled samples is enough to observe significant effects.

We show that our approach gives control over discovered properties and factors and that our method achieves better image quality. Here we provide an information theoretic underpinning shedding light on the reason for these gains. By increasing both $I(C_{ss}; X)$ and $I(C_{ss}; \tilde{X})$, the mutual information term $I(X; \tilde{X})$ is increased as well. We make the following assumptions:

$$X \leftarrow C_{ss} \rightarrow \tilde{X}, \tag{8}$$

$$I(X; \tilde{X}) = 0 \text{ initially}, \tag{9}$$

$$H(C_{ss}) = \mathtt{C}, \tag{10}$$

where $\mathtt{C}$ is a constant and → are dependency relations. Assumption (8) follows the intuition that the data is hypothesized to arise from the interaction of independent factors. While latent factors consist of z, C_{us} and C_{ss}, we abstract for the sake of simplicity. Assumption (9) formulates the initial state of our model where the synthetic data distribution P_G and the data distribution P_{data} are independent. Finally we can assume that labels follow a fixed distribution and hence have a fixed entropy $H(C_{ss})$, giving rise to (10).

We decompose $H(C_{ss})$ and reformulate $I(X; \tilde{X})$ in the following way:

$$H(C_{ss}) = I(C_{ss}; X) + I(C_{ss}; \tilde{X}) + H(C_{ss}|X, \tilde{X}) - I(C_{ss}; X; \tilde{X}), \tag{11}$$

$$I(C_{ss}; X; \tilde{X}) = I(X; \tilde{X}) - I(X; \tilde{X}|C_{ss}) \tag{12}$$

where $I(C_{ss}; X; \tilde{X})$ is the multivariate mutual information term. While pointwise MI is per definition non-negative, in the multivariate case negative values are possible if two variables are coupled via the third. By using the conditional independence assumption (8), we have

$$I(C_{ss}; X; \tilde{X}) = I(X; \tilde{X}) - I(X; \tilde{X}|C_{ss}) = I(X; \tilde{X}) \geq 0. \tag{13}$$

Thus the entropy term $H(C_{ss})$ in Eq. (11) takes the form

$$H(C_{ss}) = I(C_{ss}; X) + I(C_{ss}; \tilde{X}) + H(C_{ss}|X, \tilde{X}) - I(X; \tilde{X}) \tag{14}$$

Let Δ symbolize the change in value of a term. According to assumption (10), the following must hold:

$$\Delta_{I(C_{ss};X)} + \Delta_{I(C_{ss};\tilde{X})} + \Delta_{H(C_{ss}|X,\tilde{X})} - \Delta_{I(X;\tilde{X})} = 0 \tag{15}$$

Note that $\Delta_{I(C_{ss};X)}$ and $\Delta_{I(C_{ss};\tilde{X})}$ increase during training since we directly optimize these terms, leading to the following cases:

$$\begin{aligned} \Delta_{I(C_{ss};X)} + \Delta_{I(C_{ss};\tilde{X})} &\geq -\Delta_{H(C_{ss}|X,\tilde{X})} \implies \Delta_{I(X;\tilde{X})} \geq 0 \\ \Delta_{I(C_{ss};X)} + \Delta_{I(C_{ss};\tilde{X})} &< -\Delta_{H(C_{ss}|X,\tilde{X})} \implies \Delta_{I(X;\tilde{X})} < 0 \end{aligned} \tag{16}$$

The first case results in the desired behavior. However the latter case cannot occur, as it would result in negative mutual information $I(X; \tilde{X})$. Hence, based on our assumptions, increasing both $I(C_{ss}; X)$ and $I(C_{ss}; \tilde{X})$ leads to an increase in $I(X; \tilde{X})$.

4 Implementation

For both D and G of ss-InfoGAN we use a similar architecture with DCGAN [21], which is reported to stabilize training. The networks for the parametric distributions Q_{us} and Q_{ss} share all the layers with D except the last layers. This is similar to [3], which models Q as an extension to D. This approach has the disadvantage of negligibly higher computational cost for Q_{ss} in comparison to InfoGAN. However, this is offset by a faster convergence rate in return.

In our experiments with low amount of labeled data, we initially favor drawing labeled samples, which improves convergence rate of the supervised latent codes significantly. During training the probability of drawing a labeled sample is annealed until the actual labeled sample ratio in the data is reached. The loss function used to calculate L_I and L_{IS} is the cross-entropy for categorical latent codes and the mean squared error for continuous latent codes. The unsupervised categorical codes are sampled from a uniform categorical distribution whereas the continuous codes are sampled from a uniform distribution. All the experimental details are listed in the supplementary document. In the interest of reproducible research, we provide the source code on GitHub.[1]

For comparison we re-implement the original InfoGAN architecture in the Torch framework [4] with minor modifications. Note that there may be differences in results due to the unstable nature of GANs, possibly amplified by using a different framework and different initial conditions. In our implementation the loss function for continuous latent codes are not treated as a factored Gaussian, but approximated with the mean squared error, which leads to a slight adjustment in the architecture of Q.

5 Experiments

In our study we focus on interpretability of the representations and quality of synthetic images under different amount of labeling. The closest related work to that of ours is InfoGAN, and the aim was to directly improve upon that architecture. The existing semi-supervised generative modeling studies on the other hand, aim to learn discriminative representations for classification. Therefore we make a direct comparison with InfoGAN.

We evaluate our model on the MNIST [13], SVHN [19], CelebA [14] and CIFAR-10 [10] datasets. First, we inspect how well ss-InfoGAN learns the representations as defined by existing labels. Second, we qualitatively evaluate the representations learned by ss-InfoGAN. Finally, we analyze how much labeled data is required for the model to encode semantic meaning of the labels y via c_{ss}.

We hypothesize that the quality of the generator in class-conditional sample synthesis can be quantitatively assessed by a separate classifier trained to recognize class labels. The class labels of the synthetic samples (i.e. the class conditional inputs of the generator) are regarded as true targets and compared with the classifier's predictions. In order to prevent biased results due to the

[1] Implementation can be found at https://github.com/spurra/ss-infogan.

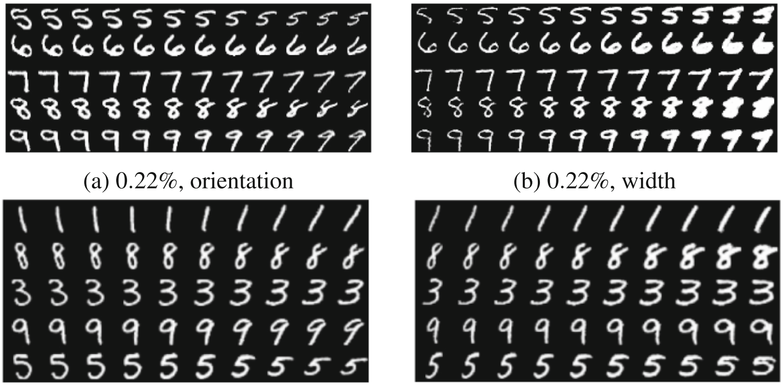
generator overfitting, we train the classifier C by using the *test set*, and validate on the *training set* for each dataset. Despite the *test set* consisting fewer samples, the classifier C generally performs well on the unseen *training set*. In our experiments, we use a standard CNN (architecture described in the supplementary file) for the MNIST, CelebA and SVHN datasets and Wide Residual Networks [28] for CIFAR-10 dataset.

In order to evaluate how well the model separates types of semantic variation, we generate synthetic images by varying only one latent factor by means of linear interpolation while keeping the remaining latent codes fixed.

To evaluate the necessary amount of supervision we perform quantitative analysis of the classifier accuracy and qualitative analysis by examining synthetic samples. To do so, we discard increasingly bigger sets of labels from the data. Note that Q_{ss} is trained only by using labeled samples and hence sees less data, whereas the rest of the architecture, namely the generator and the shared layers of the discriminator, uses the entire training samples in unsupervised manner. The minimum amount of labeled samples required to learn the representation of labels y varies depending on the dataset. However, for all our experiments it never exceeded 10%.

5.1 MNIST

MNIST is a standard dataset used to evaluate many generative models. It consists of handwritten digits, and is labeled with the digit category. Figure 2

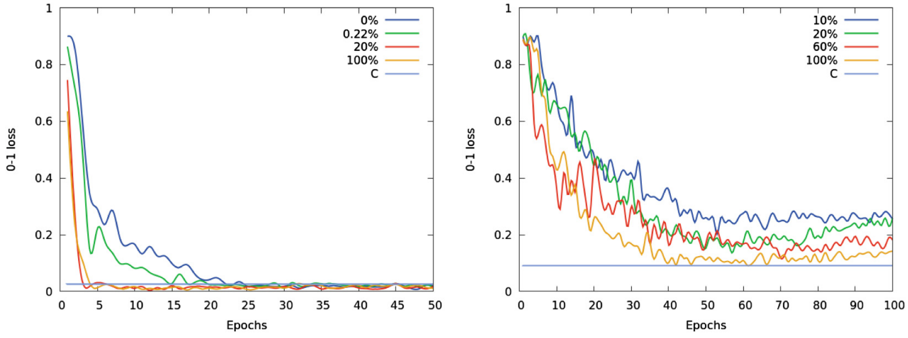

(a) 0.22%, orientation

(b) 0.22%, width

(c) 0%, orientation

(d) 0%, width

Fig. 2. Manipulating latent code on MNIST: in all figures of latent code manipulation we use the convention that a latent code varies from left to right (x-axis) while the remaining codes and the noise are kept fixed. Each row along the y-axis corresponds to a categorical latent code encoding a class label unless otherwise stated. The interpretation of the varying latent code is provided under the image. Synthetic images generated by interpolating the latent codes, encoding the digit "orientation" and "width", between −2 and 2. (a, b) ss-InfoGAN with 0.22% supervision. (c, d) InfoGAN (taken from [3]).



(a) MNIST

(b) SVHN

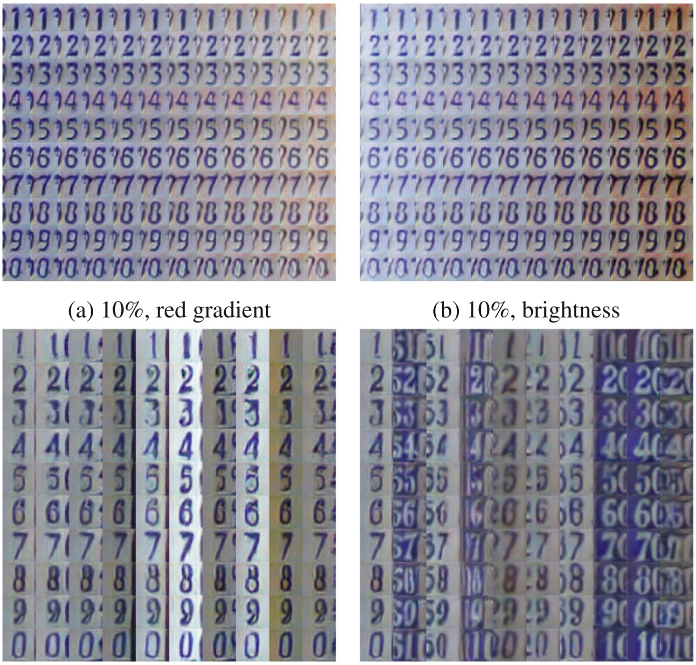
Fig. 3. 0–1 loss on synthetic samples: in all 0–1 loss figures we plot the classification accuracy on synthetic samples of the respective dataset. During training of ss-InfoGAN, a batch of synthetic samples are randomly generated, and evaluated by the independent classifier C. Colors represent the GAN models trained with different amount of supervision and classifier performance (C) on real validation samples. (Color figure online)

presents the synthetic samples generated with our model and InfoGAN by varying the latent code. Due to lower complexity of the dataset, InfoGAN is capable of learning the digit representation unsupervised. However, using just 0.22% of the available data has a two-fold benefit. First, semi-supervision provides additional fine-grained control (e.g., digits are already sorted in ascending manner in Fig. 2a, b). Second, we experimentally verified that the additional information increases convergence speed of the generator, illustrated in Fig. 3a. The 0–1 loss of the classifier C decreases faster as more labeled samples are introduced while the fully unsupervised setting (i.e. InfoGAN) is the slowest. The smallest amount of labeled samples for which the effect of supervision is observable is 0.22% of the dataset, which corresponds to 132 labeled samples out of $60'000$.

5.2 SVHN

Next, we run ss-InfoGAN on the SVHN dataset which consists of color images, hence includes more noise and natural effects such as illumination. Similar to MNIST, this dataset is labeled with respect to the digit category. In Fig. 4, latent codes with various interpretation are presented. In this experiment different amount of supervision result in different unsupervised representations retrieved.

The SVHN dataset is perturbed by various noise factors such as blur and ambiguity in the digit categories. Figure 5 compares real samples with randomly generated synthetic samples by varying digit categories. The InfoGAN configuration (0% supervision) fails to encode a categorical latent code for the digit category. Leveraging some labeled information, our model becomes more robust to perturbations in the images. Through the introduction of labeled samples we are capable of exerting control over the latent space, encoding the digit labels in



(a) 10%, red gradient

(b) 10%, brightness

(c) 100%, brightness

(d) 100%, digit font

Fig. 4. Manipulating latent code on SVHN: latent codes encoding the "brightness", "digit font" and "red gradient" are interpolated between −2 and 2 for each semi-supervised categorical code. (a, b) ss-InfoGAN with 10% supervision. (c, d) ss-InfoGAN with 100% supervision. (Color figure online)

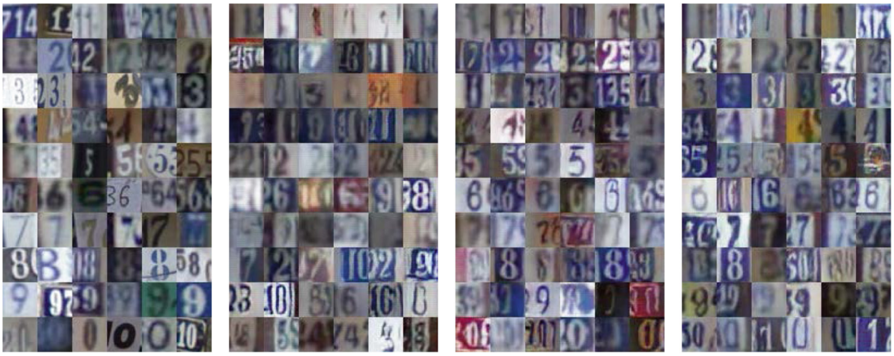
the categorical latent code c_{ss}. The smallest fraction of labeled data needed to achieve a notable effect is 10% (i.e. $7'326$ labels out of $73'257$ samples).

In Fig. 3b we assess the performance of ss-InfoGAN with respect to C. The unsupervised configuration is left out since it is not able to control digit categories. As ss-InfoGAN exploits more label information, the generator converges faster and synthesizes more accurate images in terms of digit recognizability.

5.3 CelebA

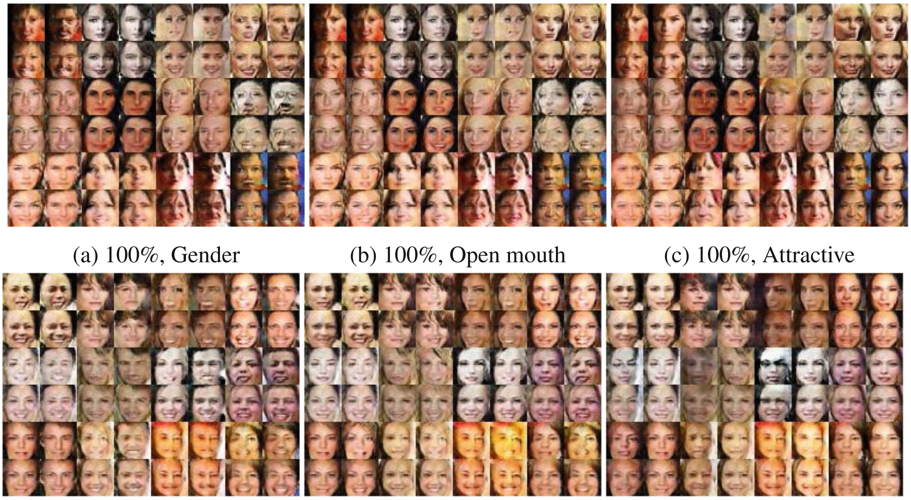
The CelebA dataset contains a rich variety of binary labels. We pre-process the data by extracting the faces via a face detector and then resize the extracted faces to 32 × 32. From the set of binary labels provided in the data we select the following attributes: "presence of smile", "mouth open", "attractive" and "gender".

Figure 6 shows synthetic images generated with ss-InfoGAN by varying certain latent codes. Although we experiment by using various hyper-parameters, InfoGAN is not able to learn an equivalent representation to these attributes.



(a) Real samples (b) 0% (c) 10% (d) 100%

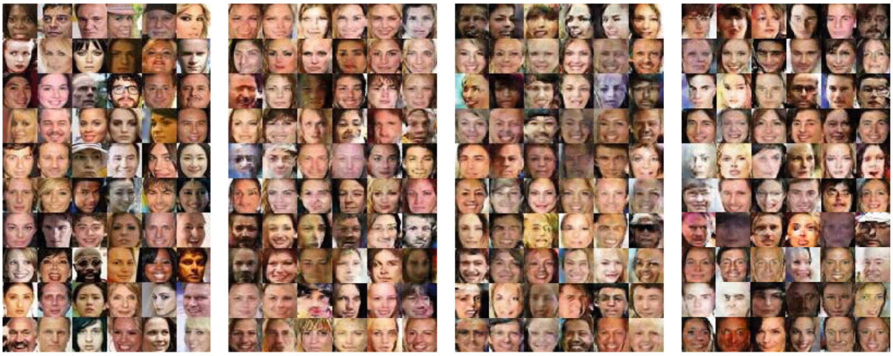
Fig. 5. Random synthetic SVHN samples: in all figures of randomly synthesized images we present examples of real samples from the dataset and synthetic images. Models are trained with different amount of supervision which is noted under the images, where the 0% supervision corresponds to InfoGAN. In each row, the semi-supervised categorical code encoding the digits is kept fixed while rest of the input vector, (i.e. z, c_{us}) and the remaining codes in c_{ss}, is randomly drawn from the latent distribution. Although each row represents a digit the fully unsupervised model (b) (i.e. InfoGAN) lacks control on the digit category.



(a) 100%, Gender (b) 100%, Open mouth (c) 100%, Attractive

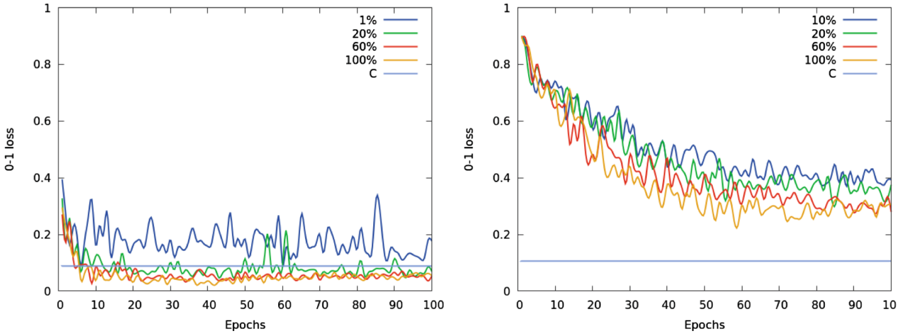
(d) 1%, Gender (e) 1%, Open mouth (f) 1%, Attractive

Fig. 6. Manipulating latent code on CelebA: synthetic samples generated by varying semi-supervised latent codes for the binary attributes "gender", "open mouth" and "attractive". Each 2 × 2 block corresponds to synthetic samples generated by keeping the input vector (c_{ss}, c_{us}, z) fixed except the first semi-supervised categorical code encoding "smile" attribute (varied across the y-axis) and the other semi-supervised categorical latent code, whose interpretation is given in the caption (varied across the x-axis).



(a) Real samples (b) 0% (c) 1% (d) 100%

Fig. 7. Random synthetic CelebA samples: for each synthesized image the latent and noise variables are randomly drawn from their respective distribution.



(a) CelebA (b) CIFAR-10

Fig. 8. 0–1 loss on synthetic samples: (a) the model trained on CelebA dataset by leveraging the minimum amount of supervision that is sufficient to encode label (1%) information shows unstable behavior. (b) None of the models trained on CIFAR-10 dataset achieve the quality enough to reach real sample classification accuracy.

We see that for as low as 1%, c_{ss} acquires the semantic meaning of y. This corresponds to $1'511$ labeled samples out of $151'162$. Figure 7 presents a batch of real samples from the dataset alongside with randomly synthesized samples from generators trained on various labeled percentages, with 0% corresponding again to InfoGAN.

The performance of ss-InfoGAN on the independent classifier C is shown in Fig. 8a. For the lowest amount of labeling some instability can be observed. We believe this is due to the differences between the positives and the negatives of each binary label being more subtle than in other datasets. In addition, synthetic data generation exhibits certain variability which can obfuscate important parts of the image. However, using 20% of labeled samples ensures a stable training performance.

5.4 CIFAR-10

Finally we evaluate our model on CIFAR-10 dataset consisting of natural images. The data is labeled with the object category, which we use for the first semi-supervised categorical code. In order to stabilize training we apply instance noise [25].

On this dataset the unsupervised latent codes are not interpretable. An example is presented in Fig. 9 where the synthetic samples are generated by varying one of the unsupervised latent codes. Despite the fact that ss-InfoGAN model is trained by using *all* label information, the semantic meaning of this unsupervised representation is not clear. The randomness of the natural images prevent models from learning interpretable representations in the absence of guidance.

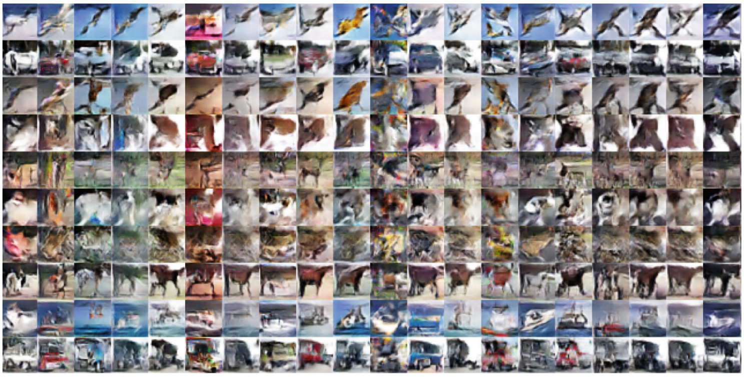
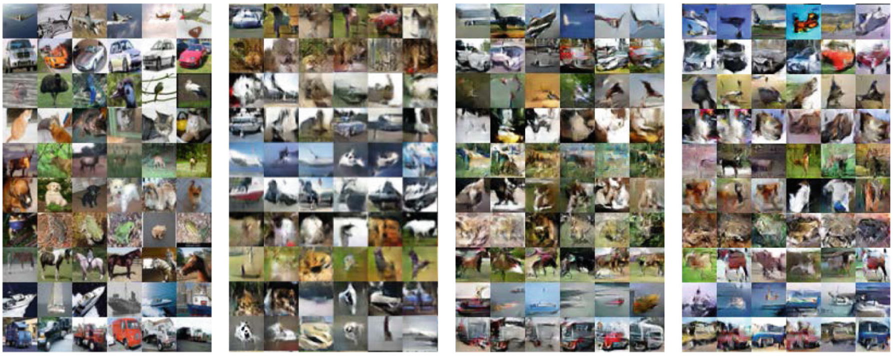


Fig. 9. Manipulating latent code on CIFAR: an example of varying an unsupervised latent code (x-axis) on CIFAR-10. Each row corresponds to a fixed code and represents class labels. The unsupervised latent code is not clearly interpretable.



(a) Real samples (b) 0% (c) 10% (d) 100%

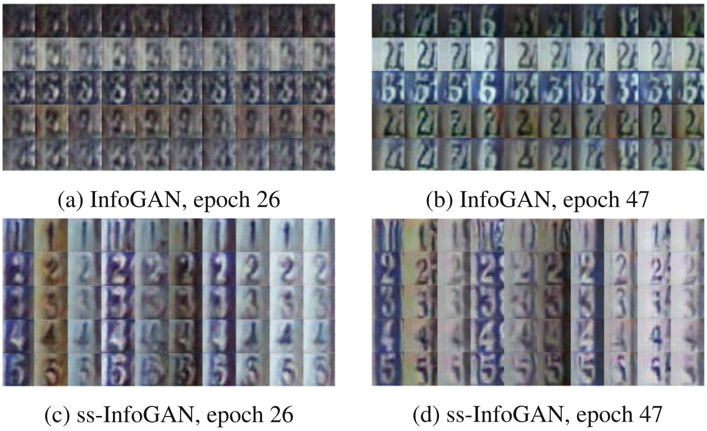
Fig. 10. Random synthetic CIFAR-10 samples: real samples and synthetic samples generated by the models trained with different amount of supervision. In each row, the semi-supervised categorical code encoding the image categories is kept fixed while rest of the input vector (i.e. z, c_{us}) and the remaining codes in c_{ss}, is randomly drawn from the latent distribution.

Figure 10 shows synthetic samples generated by models with different supervision configurations. InfoGAN has difficulties in learning the object category (see Fig. 10b) and hence in generating class-conditional synthetic images. For this dataset we find that labeling 10% of the training data (corresponding to 5′000 images out of 50′000) is sufficient for ss-InfoGAN to encode class category (see Fig. 10c).

In Fig. 8b classification accuracy of C on the synthetic samples is plotted, again displaying the similar behavior of having better performance as more labels are available. It is evident that the additional information provided by the labels is fundamental to control *what* the image depicts. We argue that attaining such low amounts of labels is feasible even for large and complicated datasets.

5.5 Convergence Speed of Sample Quality

During the course of the experiments, it is observed that the convergence of synthetic sample quality is faster in comparison to InfoGAN. Figure 11 shows synthetic SVHN samples from a fully supervised ss-InfoGAN and InfoGAN at training epoch 26 and 47. The training epochs are chosen by inspection so that each model starts producing recognizable images. Therefore we can quantitatively say that ss-InfoGAN converges faster than InfoGAN.


(a) InfoGAN, epoch 26

(b) InfoGAN, epoch 47

(c) ss-InfoGAN, epoch 26

(d) ss-InfoGAN, epoch 47

Fig. 11. Samples from InfoGAN and ss-InfoGAN trained on SVHN at two different epochs

6 Conclusion

We have introduced ss-InfoGAN a novel semi-supervised generative model. We have shown that including few labels increases the convergence speed of the latent codes c_{ss} and that these represent the same meaning as the labels y. This speed-up increases as more data samples are labeled. Although in theory this only improves convergence speed of c_{ss}, we have shown empirically that the sample quality convergence speed has improved as well.

In addition, it was shown that using labeling information is useful in cases where InfoGAN fails to find a *specific* representation, such as in the case of SVHN, CelebA and CIFAR-10. To successfully guide a latent code to the desired representation, it is sufficient that the dataset contains only a minimal subset of labeled data. The amount of required labels ranges from 0.22% for the simplest datasets (MNIST) to a maximum of 10% for the most complex datasets (CIFAR-10). We argue that acquiring such low percentages of labels is cost effective and makes the proposed architecture an attractive choice if control over *specific* latent codes is required and full supervision is not an option.

Acknowledgements. This work was supported in parts by the ERC grant OPTINT (StG-2016-717054).

References

1. Barber, D., Agakov, F.V.: The IM algorithm: a variational approach to information maximization. In: NIPS, pp. 201–208 (2003)
2. Burda, Y., Grosse, R., Salakhutdinov, R.: Importance weighted autoencoders. arXiv preprint arXiv:1509.00519 (2015)
3. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: InfoGAN: interpretable representation learning by information maximizing generative adversarial nets. ArXiv e-prints, June 2016
4. Collobert, R., Kavukcuoglu, K., Farabet, C.: Torch7: a matlab-like environment for machine learning. In: BigLearn, NIPS Workshop (2011)
5. Desjardins, G., Courville, A., Bengio, Y.: Disentangling factors of variation via generative entangling. ArXiv e-prints, October 2012
6. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. ArXiv e-prints, June 2014
7. Kingma, D.P., Rezende, D.J., Mohamed, S., Welling, M.: Semi-supervised learning with deep generative models. ArXiv e-prints, June 2014
8. Kingma, D.P., Salimans, T., Welling, M.: Improving variational inference with inverse autoregressive flow. arXiv preprint arXiv:1606.04934 (2016)
9. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
10. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Technical report, University of Toronto (2009)
11. Kulkarni, T.D., Whitney, W., Kohli, P., Tenenbaum, J.B.: Deep convolutional inverse graphics network. ArXiv e-prints, March 2015
12. Lamb, A., Dumoulin, V., Courville, A.: Discriminative regularization for generative models. arXiv preprint arXiv:1602.03220 (2016)
13. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
14. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV), December 2015

15. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial autoencoders. ArXiv e-prints, November 2015
16. Mathieu, M.F., Zhao, J.J., Zhao, J., Ramesh, A., Sprechmann, P., LeCun, Y.: Disentangling factors of variation in deep representation using adversarial training. In: Advances in Neural Information Processing Systems, pp. 5041–5049 (2016)
17. Metz, L., Poole, B., Pfau, D., Sohl-Dickstein, J.: Unrolled generative adversarial networks. arXiv preprint arXiv:1611.02163 (2016)
18. Mirza, M., Osindero, S.: Conditional generative adversarial nets. ArXiv e-prints, November 2014
19. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011 (2011)
20. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier GANs. ArXiv e-prints, October 2016
21. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. ArXiv e-prints, November 2015
22. Reed, S., Sohn, K., Zhang, Y., Lee, H.: Learning to disentangle factors of variation with manifold interaction. In: Proceedings of the 31st International Conference on Machine Learning (ICML-2014), pp. 1431–1439 (2014)
23. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems, pp. 2226–2234 (2016)
24. Siddharth, N., Paige, B., Van de Meent, J.W., Desmaison, A., Wood, F., Goodman, N.D., Kohli, P., Torr, P.H.S.: Learning disentangled representations with semi-supervised deep generative models. ArXiv e-prints, June 2017
25. Sønderby, C.K., Caballero, J., Theis, L., Shi, W., Huszár, F.: Amortised MAP inference for image super-resolution. CoRR abs/1610.04490 (2016). http://arxiv.org/abs/1610.04490
26. Sønderby, C.K., Raiko, T., Maaløe, L., Sønderby, S.K., Winther, O.: Ladder variational autoencoders. In: Advances in Neural Information Processing Systems, pp. 3738–3746 (2016)
27. Tenenbaum, J.B., Freeman, W.T.: Separating style and content with bilinear models. Neural Comput. **12**(6), 1247–1283 (2000). https://doi.org/10.1162/089976600300015349
28. Zagoruyko, S., Komodakis, N.: Wide residual networks. CoRR abs/1605.07146 (2016). http://arxiv.org/abs/1605.07146
29. Zhu, Z., Luo, P., Wang, X., Tang, X.: Deep learning multi-view representation for face recognition. ArXiv e-prints, June 2014