ADVERSARIAL FEATURE LEARNING

Jeff Donahue
jdonahue@cs.berkeley.edu
Computer Science Division
University of California, Berkeley

Philipp Krähenbühl
philkr@utexas.edu
Department of Computer Science
University of Texas, Austin

Trevor Darrell
trevor@eecs.berkeley.edu
Computer Science Division
University of California, Berkeley

ABSTRACT

The ability of the Generative Adversarial Networks (GANs) framework to learn generative models mapping from simple latent distributions to arbitrarily complex data distributions has been demonstrated empirically, with compelling results showing that the latent space of such generators captures semantic variation in the data distribution. Intuitively, models trained to predict these semantic latent representations given data may serve as useful feature representations for auxiliary problems where semantics are relevant. However, in their existing form, GANs have no means of learning the inverse mapping – projecting data back into the latent space. We propose Bidirectional Generative Adversarial Networks (BiGANs) as a means of learning this inverse mapping, and demonstrate that the resulting learned feature representation is useful for auxiliary supervised discrimination tasks, competitive with contemporary approaches to unsupervised and self-supervised feature learning.

1 INTRODUCTION

Deep convolutional networks (convnets) have become a staple of the modern computer vision pipeline. After training these models on a massive database of image-label pairs like ImageNet (Russakovsky et al., 2015), the network easily adapts to a variety of similar visual tasks, achieving impressive results on image classification (Donahue et al., 2014; Zeiler & Fergus, 2014; Razavian et al., 2014) or localization (Girshick et al., 2014; Long et al., 2015) tasks. In other perceptual domains such as natural language processing or speech recognition, deep networks have proven highly effective as well (Bahdanau et al., 2015; Sutskever et al., 2014; Vinyals et al., 2015; Graves et al., 2013). However, all of these recent results rely on a supervisory signal from large-scale databases of hand-labeled data, ignoring much of the useful information present in the structure of the data itself.

Meanwhile, Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) have emerged as a powerful framework for learning generative models of arbitrarily complex data distributions. The GAN framework learns a *generator* mapping samples from an arbitrary latent distribution to data, as well as an adversarial *discriminator* which tries to distinguish between real and generated samples as accurately as possible. The generator's goal is to "fool" the discriminator by producing samples which are as close to real data as possible. When trained on databases of natural images, GANs produce impressive results (Radford et al., 2016; Denton et al., 2015).

Interpolations in the latent space of the generator produce smooth and plausible semantic variations, and certain directions in this space correspond to particular semantic attributes along which the data distribution varies. For example, Radford et al. (2016) showed that a GAN trained on a database of human faces learns to associate particular latent directions with gender and the presence of eyeglasses.

A natural question arises from this ostensible "semantic juice" flowing through the weights of generators learned using the GAN framework: can GANs be used for unsupervised learning of rich feature representations for arbitrary data distributions? An obvious issue with doing so is that the

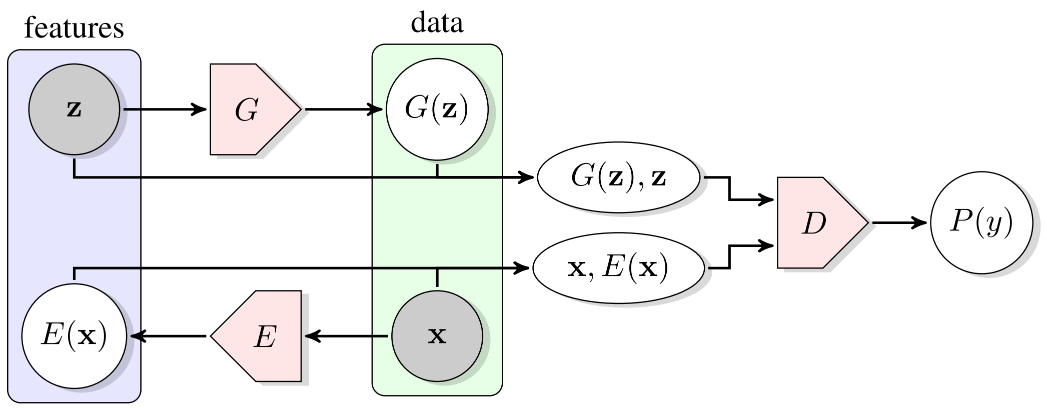

Figure 1: The structure of Bidirectional Generative Adversarial Networks (BiGAN).

generator maps latent samples to generated data, but the framework does not include an *inverse* mapping from data to latent representation.

Hence, we propose a novel unsupervised feature learning framework, *Bidirectional Generative Adversarial Networks* (BiGAN). The overall model is depicted in Figure 1. In short, in addition to the generator G from the standard GAN framework (Goodfellow et al., 2014), BiGAN includes an *encoder* E which maps data $\mathbf{x}$ to latent representations $\mathbf{z}$. The BiGAN discriminator D discriminates not only in data space ($\mathbf{x}$ versus $G(\mathbf{z})$), but jointly in data and latent space (tuples $(\mathbf{x}, E(\mathbf{x}))$ versus $(G(\mathbf{z}), \mathbf{z})$), where the latent component is either an encoder output $E(\mathbf{x})$ or a generator input $\mathbf{z}$.

It may not be obvious from this description that the BiGAN encoder E should learn to invert the generator G. The two modules cannot directly "communicate" with one another: the encoder never "sees" generator outputs ($E(G(\mathbf{z}))$ is not computed), and vice versa. Yet, in Section 3, we will both argue intuitively and formally prove that the encoder and generator must learn to invert one another in order to fool the BiGAN discriminator.

Because the BiGAN encoder learns to predict features $\mathbf{z}$ given data $\mathbf{x}$, and prior work on GANs has demonstrated that these features capture semantic attributes of the data, we hypothesize that a trained BiGAN encoder may serve as a useful feature representation for related semantic tasks, in the same way that fully supervised visual models trained to predict semantic "labels" given images serve as powerful feature representations for related visual tasks. In this context, a latent representation $\mathbf{z}$ may be thought of as a "label" for $\mathbf{x}$, but one which came for "free," without the need for supervision.

An alternative approach to learning the inverse mapping from data to latent representation is to directly model $p(\mathbf{z}|G(\mathbf{z}))$, predicting generator input $\mathbf{z}$ given generated data $G(\mathbf{z})$. We'll refer to this alternative as a *latent regressor*, later arguing (Section 4.1) that the BiGAN encoder may be preferable in a feature learning context, as well as comparing the approaches empirically.

BiGANs are a robust and highly generic approach to unsupervised feature learning, making no assumptions about the structure or type of data to which they are applied, as our theoretical results will demonstrate. Our empirical studies will show that despite their generality, BiGANs are competitive with contemporary approaches to self-supervised and weakly supervised feature learning designed specifically for a notoriously complex data distribution – natural images.

Dumoulin et al. (2016) independently proposed an identical model in their concurrent work, exploring the case of a stochastic encoder E and the ability of such models to learn in a semi-supervised setting.

2 Preliminaries

Let $p_{\mathbf{X}}(\mathbf{x})$ be the distribution of our data for $\mathbf{x} \in \Omega_{\mathbf{X}}$ (e.g. natural images). The goal of generative modeling is capture this data distribution using a probabilistic model. Unfortunately, exact modeling of this probability density function is computationally intractable (Hinton et al., 2006; Salakhutdinov & Hinton, 2009) for all but the most trivial models. Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) instead model the data distribution as a transformation of a fixed latent distribution $p_{\mathbf{Z}}(\mathbf{z})$ for $\mathbf{z} \in \Omega_{\mathbf{Z}}$. This transformation, called a *generator*, is expressed as a deterministic feed forward network $G : \Omega_{\mathbf{Z}} \to \Omega_{\mathbf{X}}$ with $p_G(\mathbf{x}|\mathbf{z}) = \delta\left(\mathbf{x} - G(\mathbf{z})\right)$ and $p_G(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{Z}}}\left[p_G(\mathbf{x}|\mathbf{z})\right]$. The goal is to train a generator such that $p_G(\mathbf{x}) \approx p_{\mathbf{X}}(\mathbf{x})$.

The GAN framework trains a generator, such that no discriminative model $D : \Omega_{\mathbf{X}} \mapsto [0,1]$ can distinguish samples of the data distribution from samples of the generative distribution. Both generator and discriminator are learned using the adversarial (minimax) objective $\min\limits_{G}\max\limits_{D} V(D,G)$, where

$$V(D,G) := \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{X}}}\left[\log D(\mathbf{x})\right] + \underbrace{\mathbb{E}_{\mathbf{x}\sim p_G}\left[\log\left(1 - D(\mathbf{x})\right)\right]}_{\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{Z}}}\left[\log(1-D(G(\mathbf{z})))\right]} \tag{1}$$

Goodfellow et al. (2014) showed that for an ideal discriminator the objective $C(G) := \max_D V(D,G)$ is equivalent to the Jensen-Shannon divergence between the two distributions p_G and $p_{\mathbf{X}}$.

The adversarial objective 1 does not directly lend itself to an efficient optimization, as each step in the generator G requires a full discriminator D to be learned. Furthermore, a perfect discriminator no longer provides any gradient information to the generator, as the gradient of any global or local maximum of $V(D,G)$ is 0. To provide a strong gradient signal nonetheless, Goodfellow et al. (2014) slightly alter the objective between generator and discriminator updates, while keeping the same fixed point characteristics. They also propose to optimize (1) using an alternating optimization switching between updates to the generator and discriminator. While this optimization is not guaranteed to converge, empirically it works well if the discriminator and generator are well balanced.

Despite the empirical strength of GANs as generative models of arbitrary data distributions, it is not clear how they can be applied as an unsupervised feature representation. One possibility for learning such representations is to learn an inverse mapping regressing from generated data $G(\mathbf{z})$ back to the latent input $\mathbf{z}$. However, unless the generator perfectly models the data distribution $p_{\mathbf{X}}$, a nearly impossible objective for a complex data distribution such as that of high-resolution natural images, this idea may prove insufficient.

3 Bidirectional Generative Adversarial Networks

In Bidirectional Generative Adversarial Networks (BiGANs) we not only train a generator, but additionally train an encoder $E : \Omega_{\mathbf{X}} \to \Omega_{\mathbf{Z}}$. The encoder induces a distribution $p_E(\mathbf{z}|\mathbf{x}) = \delta(\mathbf{z} - E(\mathbf{x}))$ mapping data points $\mathbf{x}$ into the latent feature space of the generative model. The discriminator is also modified to take input from the latent space, predicting $P_D(Y|\mathbf{x},\mathbf{z})$, where $Y = 1$ if $\mathbf{x}$ is real (sampled from the real data distribution $p_{\mathbf{X}}$), and $Y = 0$ if $\mathbf{x}$ is generated (the output of $G(\mathbf{z}), \mathbf{z} \sim p_{\mathbf{Z}}$).

The BiGAN training objective is defined as a minimax objective

$$\min_{G,E}\max_{D} V(D,E,G) \tag{2}$$

where

$$V(D,E,G) := \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{X}}}\Big[\underbrace{\mathbb{E}_{\mathbf{z}\sim p_E(\cdot|\mathbf{x})}\left[\log D(\mathbf{x},\mathbf{z})\right]}_{\log D(\mathbf{x},E(\mathbf{x}))}\Big] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{Z}}}\Big[\underbrace{\mathbb{E}_{\mathbf{x}\sim p_G(\cdot|\mathbf{z})}\left[\log\left(1 - D(\mathbf{x},\mathbf{z})\right)\right]}_{\log(1-D(G(\mathbf{z}),\mathbf{z}))}\Big]. \tag{3}$$

We optimize this minimax objective using the same alternating gradient based optimization as Goodfellow et al. (2014). See Section 3.4 for details.

BiGANs share many of the theoretical properties of GANs (Goodfellow et al., 2014), while additionally guaranteeing that at the global optimum, G and E are each other's inverse. BiGANs are also closely related to autoencoders with an ℓ_0 loss function. In the following sections we highlight some of the appealing theoretical properties of BiGANs.

Definitions Let $p_{G\mathbf{Z}}(\mathbf{x},\mathbf{z}) := p_G(\mathbf{x}|\mathbf{z})p_{\mathbf{Z}}(\mathbf{z})$ and $p_{E\mathbf{X}}(\mathbf{x},\mathbf{z}) := p_E(\mathbf{z}|\mathbf{x})p_{\mathbf{X}}(\mathbf{x})$ be the joint distributions modeled by the generator and encoder respectively. $\Omega := \Omega_{\mathbf{X}} \times \Omega_{\mathbf{Z}}$ is the joint latent and data space. For a region $R \subseteq \Omega$,

$$P_{E\mathbf{X}}(R) := \int_\Omega p_{E\mathbf{X}}(\mathbf{x},\mathbf{z})\mathbf{1}_{[(\mathbf{x},\mathbf{z})\in R]}\,\mathrm{d}(\mathbf{x},\mathbf{z}) = \int_{\Omega_\mathbf{X}} p_\mathbf{X}(\mathbf{x}) \int_{\Omega_\mathbf{Z}} p_E(\mathbf{z}|\mathbf{x})\mathbf{1}_{[(\mathbf{x},\mathbf{z})\in R]}\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{x}$$
$$P_{G\mathbf{Z}}(R) := \int_\Omega p_{G\mathbf{Z}}(\mathbf{x},\mathbf{z})\mathbf{1}_{[(\mathbf{x},\mathbf{z})\in R]}\,\mathrm{d}(\mathbf{x},\mathbf{z}) = \int_{\Omega_\mathbf{Z}} p_\mathbf{Z}(\mathbf{z}) \int_{\Omega_\mathbf{X}} p_G(\mathbf{x}|\mathbf{z})\mathbf{1}_{[(\mathbf{x},\mathbf{z})\in R]}\,\mathrm{d}\mathbf{x}\,\mathrm{d}\mathbf{z}$$

are probability measures over that region. We also define

$$P_\mathbf{X}(R_\mathbf{X}) := \int_{\Omega_\mathbf{X}} p_\mathbf{X}(\mathbf{x})\mathbf{1}_{[\mathbf{x}\in R_\mathbf{X}]}\,\mathrm{d}\mathbf{x} \qquad P_\mathbf{Z}(R_\mathbf{Z}) := \int_{\Omega_\mathbf{Z}} p_\mathbf{Z}(\mathbf{z})\mathbf{1}_{[\mathbf{z}\in R_\mathbf{Z}]}\,\mathrm{d}\mathbf{z}$$

as measures over regions $R_\mathbf{X} \subseteq \Omega_\mathbf{X}$ and $R_\mathbf{Z} \subseteq \Omega_\mathbf{Z}$. We refer to the set of features and data samples in the support of $P_\mathbf{X}$ and $P_\mathbf{Z}$ as $\hat{\Omega}_\mathbf{X} := \mathrm{supp}(P_\mathbf{X})$ and $\hat{\Omega}_\mathbf{Z} := \mathrm{supp}(P_\mathbf{Z})$ respectively. $\mathrm{D_{KL}}\left(P \,||\, Q\right)$ and $\mathrm{D_{JS}}\left(P \,||\, Q\right)$ respectively denote the Kullback-Leibler (KL) and Jensen-Shannon divergences between probability measures P and Q. By definition,

$$\mathrm{D_{KL}}\left(P \,||\, Q\right) := \mathbb{E}_{\mathbf{x}\sim P}\left[\log f_{PQ}(\mathbf{x})\right]$$
$$\mathrm{D_{JS}}\left(P \,||\, Q\right) := \tfrac{1}{2}\left(\mathrm{D_{KL}}\left(P \,\middle|\middle|\, \tfrac{P+Q}{2}\right) + \mathrm{D_{KL}}\left(Q \,\middle|\middle|\, \tfrac{P+Q}{2}\right)\right),$$

where $f_{PQ} := \frac{\mathrm{d}P}{\mathrm{d}Q}$ is the Radon-Nikodym (RN) derivative of measure P with respect to measure Q, with the defining property that $P(R) = \int_R f_{PQ}\,\mathrm{d}Q$. The RN derivative $f_{PQ} : \Omega \mapsto \mathbb{R}_{\geq 0}$ is defined for any measures P and Q on space Ω such that P is absolutely continuous with respect to Q: i.e., for any $R \subseteq \Omega$, $P(R) > 0 \implies Q(R) > 0$.

3.1 Optimal discriminator, generator, & encoder

We start by characterizing the optimal discriminator for any generator and encoder, following Goodfellow et al. (2014). This optimal discriminator then allows us to reformulate objective (3), and show that it reduces to the Jensen-Shannon divergence between the joint distributions $P_{E\mathbf{X}}$ and $P_{G\mathbf{Z}}$.

Proposition 1 *For any E and G, the optimal discriminator $D^*_{EG} := \arg\max_D V(D,E,G)$ is the Radon-Nikodym derivative $f_{EG} := \frac{\mathrm{d}P_{E\mathbf{X}}}{\mathrm{d}(P_{E\mathbf{X}}+P_{G\mathbf{Z}})} : \Omega \mapsto [0,1]$ of measure $P_{E\mathbf{X}}$ with respect to measure $P_{E\mathbf{X}} + P_{G\mathbf{Z}}$.*

Proof. Given in Appendix A.1.

This optimal discriminator now allows us to characterize the optimal generator and encoder.

Proposition 2 *The encoder and generator's objective for an optimal discriminator $C(E,G) := \max_D V(D,E,G) = V(D^*_{EG},E,G)$ can be rewritten in terms of the Jensen-Shannon divergence between measures $P_{E\mathbf{X}}$ and $P_{G\mathbf{Z}}$ as $C(E,G) = 2\,\mathrm{D_{JS}}\left(P_{E\mathbf{X}} \,||\, P_{G\mathbf{Z}}\right) - \log 4$.*

Proof. Given in Appendix A.2.

Theorem 1 *The global minimum of $C(E,G)$ is achieved if and only if $P_{E\mathbf{X}} = P_{G\mathbf{Z}}$. At that point, $C(E,G) = -\log 4$ and $D^*_{EG} = \frac{1}{2}$.*

Proof. From Proposition 2, we have that $C(E,G) = 2\,\mathrm{D_{JS}}\left(P_{E\mathbf{X}} \,||\, P_{G\mathbf{Z}}\right) - \log 4$. The Jensen-Shannon divergence $\mathrm{D_{JS}}\left(P \,||\, Q\right) \geq 0$ for any P and Q, and $\mathrm{D_{JS}}\left(P \,||\, Q\right) = 0$ if and only if $P = Q$. Therefore, the global minimum of $C(E,G)$ occurs if and only if $P_{E\mathbf{X}} = P_{G\mathbf{Z}}$, and at this point the value is $C(E,G) = -\log 4$. Finally, $P_{E\mathbf{X}} = P_{G\mathbf{Z}}$ implies that the optimal discriminator is chance: $D^*_{EG} = \frac{\mathrm{d}P_{E\mathbf{X}}}{\mathrm{d}(P_{E\mathbf{X}}+P_{G\mathbf{Z}})} = \frac{\mathrm{d}P_{E\mathbf{X}}}{2\,\mathrm{d}P_{E\mathbf{X}}} = \frac{1}{2}$. □

The optimal discriminator, encoder, and generator of BiGAN are similar to the optimal discriminator and generator of the GAN framework (Goodfellow et al., 2014). However, an important difference is that BiGAN optimizes a Jensen-Shannon divergence between a joint distribution over both data $\mathbf{X}$ and latent features $\mathbf{Z}$. This joint divergence allows us to further characterize properties of G and E, as shown below.

3.2 Optimal generator & encoder are inverses

We first present an intuitive argument that, in order to "fool" a perfect discriminator, a deterministic BiGAN encoder and generator must invert each other. (Later we will formally state and prove this property.) Consider a BiGAN discriminator input pair $(\mathbf{x}, \mathbf{z})$. Due to the sampling procedure, $(\mathbf{x}, \mathbf{z})$ must satisfy at least one of the following two properties:

$$\text{(a) } \mathbf{x} \in \hat{\Omega}_{\mathbf{X}} \wedge E(\mathbf{x}) = \mathbf{z} \qquad\qquad \text{(b) } \mathbf{z} \in \hat{\Omega}_{\mathbf{Z}} \wedge G(\mathbf{z}) = \mathbf{x}$$

If *only* one of these properties is satisfied, a perfect discriminator can infer the source of $(\mathbf{x}, \mathbf{z})$ with certainty: if only (a) is satisfied, $(\mathbf{x}, \mathbf{z})$ must be an encoder pair $(\mathbf{x}, E(\mathbf{x}))$ and $D^*_{EG}(\mathbf{x}, \mathbf{z}) = 1$; if only (b) is satisfied, $(\mathbf{x}, \mathbf{z})$ must be a generator pair $(G(\mathbf{z}), \mathbf{z})$ and $D^*_{EG}(\mathbf{x}, \mathbf{z}) = 0$.

Therefore, in order to fool a perfect discriminator at $(\mathbf{x}, \mathbf{z})$ (so that $0 < D^*_{EG}(\mathbf{x}, \mathbf{z}) < 1$), E and G must satisfy *both* (a) and (b). In this case, we can substitute the equality $E(\mathbf{x}) = \mathbf{z}$ required by (a) into the equality $G(\mathbf{z}) = \mathbf{x}$ required by (b), and vice versa, giving the inversion properties $\mathbf{x} = G(E(\mathbf{x}))$ and $\mathbf{z} = E(G(\mathbf{z}))$.

Formally, we show in Theorem 2 that the optimal generator and encoder invert one another almost everywhere on the support $\hat{\Omega}_{\mathbf{X}}$ and $\hat{\Omega}_{\mathbf{Z}}$ of $P_{\mathbf{X}}$ and $P_{\mathbf{Z}}$.

Theorem 2 *If E and G are an optimal encoder and generator, then $E = G^{-1}$ almost everywhere; that is, $G(E(\mathbf{x})) = \mathbf{x}$ for $P_{\mathbf{X}}$-almost every $\mathbf{x} \in \Omega_{\mathbf{X}}$, and $E(G(\mathbf{z})) = \mathbf{z}$ for $P_{\mathbf{Z}}$-almost every $\mathbf{z} \in \Omega_{\mathbf{Z}}$.*

Proof. Given in Appendix A.4.

While Theorem 2 characterizes the encoder and decoder at their optimum, due to the non-convex nature of the optimization, this optimum might never be reached. Experimentally, Section 4 shows that on standard datasets, the two are approximate inverses; however, they are rarely exact inverses. It is thus also interesting to show what objective BiGAN optimizes in terms of E and G. Next we show that BiGANs are closely related to autoencoders with an ℓ_0 loss function.

3.3 Relationship to autoencoders

As argued in Section 1, a model trained to predict features $\mathbf{z}$ given data $\mathbf{x}$ should learn useful semantic representations. Here we show that the BiGAN objective forces the encoder E to do exactly this: in order to fool the discriminator at a particular $\mathbf{z}$, the encoder must invert the generator at that $\mathbf{z}$, such that $E(G(\mathbf{z})) = \mathbf{z}$.

Theorem 3 *The encoder and generator objective given an optimal discriminator $C(E, G) := \max_D V(D, E, G)$ can be rewritten as an ℓ_0 autoencoder loss function*

$$\begin{aligned} C(E,G) = \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{X}}} & \left[\mathbf{1}_{\left[E(\mathbf{x})\in\hat{\Omega}_{\mathbf{Z}} \wedge G(E(\mathbf{x}))=\mathbf{x}\right]} \log f_{EG}(\mathbf{x}, E(\mathbf{x})) \right] + \\ & \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{Z}}} \left[\mathbf{1}_{\left[G(\mathbf{z})\in\hat{\Omega}_{\mathbf{X}} \wedge E(G(\mathbf{z}))=\mathbf{z}\right]} \log\left(1 - f_{EG}(G(\mathbf{z}), \mathbf{z})\right) \right] \end{aligned}$$

with $\log f_{EG} \in (-\infty, 0)$ *and* $\log(1 - f_{EG}) \in (-\infty, 0)$ *$P_{E\mathbf{X}}$-almost and $P_{G\mathbf{Z}}$-almost everywhere.*

Proof. Given in Appendix A.5.

Here the indicator function $\mathbf{1}_{[G(E(\mathbf{x}))=\mathbf{x}]}$ in the first term is equivalent to an autoencoder with ℓ_0 loss, while the indicator $\mathbf{1}_{[E(G(\mathbf{z}))=\mathbf{z}]}$ in the second term shows that the BiGAN encoder must invert the generator, the desired property for feature learning. The objective further encourages the functions $E(\mathbf{x})$ and $G(\mathbf{z})$ to produce valid outputs in the support of $P_{\mathbf{Z}}$ and $P_{\mathbf{X}}$ respectively. Unlike regular autoencoders, the ℓ_0 loss function does not make any assumptions about the structure or distribution of the data itself; in fact, all the structural properties of BiGAN are learned as part of the discriminator.

3.4 Learning

In practice, as in the GAN framework (Goodfellow et al., 2014), each BiGAN module D, G, and E is a parametric function (with parameters θ_D, θ_G, and θ_E, respectively). As a whole, BiGAN can be optimized using alternating stochastic gradient steps. In one iteration, the discriminator parameters θ_D are updated by taking one or more steps in the positive gradient direction $\nabla_{\theta_D} V(D, E, G)$, then the encoder parameters θ_E and generator parameters θ_G are together updated by taking a step in the negative gradient direction $-\nabla_{\theta_E, \theta_G} V(D, E, G)$. In both cases, the expectation terms of

$V(D, E, G)$ are estimated using mini-batches of n samples $\{\mathbf{x}^{(i)} \sim p_{\mathbf{X}}\}_{i=1}^{n}$ and $\{\mathbf{z}^{(i)} \sim p_{\mathbf{Z}}\}_{i=1}^{n}$ drawn independently for each update step.

Goodfellow et al. (2014) found that an objective in which the real and generated labels Y are swapped provides stronger gradient signal to G. We similarly observed in BiGAN training that an "inverse" objective provides stronger gradient signal to G and E. For efficiency, we also update all modules D, G, and E simultaneously at each iteration, rather than alternating between D updates and G, E updates. See Appendix B for details.

3.5 Generalized BiGAN

It is often useful to parametrize the output of the generator G and encoder E in a different, usually smaller, space $\Omega'_{\mathbf{X}}$ and $\Omega'_{\mathbf{Z}}$ rather than the original $\Omega_{\mathbf{X}}$ and $\Omega_{\mathbf{Z}}$. For example, for visual feature learning, the images input to the encoder should be of similar resolution to images used in the evaluation. On the other hand, generating high resolution images remains difficult for current generative models. In this situation, the encoder may take higher resolution input while the generator output and discriminator input remain low resolution.

We generalize the BiGAN objective $V(D, G, E)$ (3) with functions $g_{\mathbf{X}} : \Omega_{\mathbf{X}} \mapsto \Omega'_{\mathbf{X}}$ and $g_{\mathbf{Z}} : \Omega_{\mathbf{Z}} \mapsto \Omega'_{\mathbf{Z}}$, and encoder $E : \Omega_{\mathbf{X}} \mapsto \Omega'_{\mathbf{Z}}$, generator $G : \Omega_{\mathbf{Z}} \mapsto \Omega'_{\mathbf{X}}$, and discriminator $D : \Omega'_{\mathbf{X}} \times \Omega'_{\mathbf{Z}} \mapsto [0, 1]$:

$$\mathbb{E}_{\mathbf{x} \sim p_{\mathbf{X}}} \Big[\underbrace{\mathbb{E}_{\mathbf{z}' \sim p_E(\cdot|\mathbf{x})} \left[\log D(g_{\mathbf{X}}(\mathbf{x}), \mathbf{z}') \right]}_{\log D(g_{\mathbf{X}}(\mathbf{x}), E(\mathbf{x}))} \Big] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}} \Big[\underbrace{\mathbb{E}_{\mathbf{x}' \sim p_G(\cdot|\mathbf{z})} \left[\log \left(1 - D(\mathbf{x}', g_{\mathbf{Z}}(\mathbf{z}))\right) \right]}_{\log(1 - D(G(\mathbf{z}), g_{\mathbf{Z}}(\mathbf{z})))} \Big]$$

An identity $g_{\mathbf{X}}(\mathbf{x}) = \mathbf{x}$ and $g_{\mathbf{Z}}(\mathbf{z}) = \mathbf{z}$ (and $\Omega'_{\mathbf{X}} = \Omega_{\mathbf{X}}$, $\Omega'_{\mathbf{Z}} = \Omega_{\mathbf{Z}}$) yields the original objective. For visual feature learning with higher resolution encoder inputs, $g_{\mathbf{X}}$ is an image resizing function that downsamples a high resolution image $\mathbf{x} \in \Omega_{\mathbf{X}}$ to a lower resolution image $\mathbf{x}' \in \Omega'_{\mathbf{X}}$, as output by the generator. ($g_{\mathbf{Z}}$ is identity.)

In this case, the encoder and generator respectively induce probability measures $P_{E\mathbf{X}'}$ and $P_{G\mathbf{Z}'}$ over regions $R \subseteq \Omega'$ of the joint space $\Omega' := \Omega'_{\mathbf{X}} \times \Omega'_{\mathbf{Z}}$, with $P_{E\mathbf{X}'}(R) := \int_{\Omega_{\mathbf{X}}} \int_{\Omega'_{\mathbf{X}}} \int_{\Omega'_{\mathbf{Z}}} p_{E\mathbf{X}}(\mathbf{x}, \mathbf{z}') \mathbf{1}_{[(\mathbf{x}', \mathbf{z}') \in R]} \delta(g_{\mathbf{X}}(\mathbf{x}) - \mathbf{x}') \, \mathrm{d}\mathbf{z}' \, \mathrm{d}\mathbf{x}' \, \mathrm{d}\mathbf{x} = \int_{\Omega_{\mathbf{X}}} p_{\mathbf{X}}(\mathbf{x}) \mathbf{1}_{[(g_{\mathbf{X}}(\mathbf{x}), E(\mathbf{x})) \in R]} \, \mathrm{d}\mathbf{x}$, and $P_{G\mathbf{Z}'}$ defined analogously. For optimal E and G, we can show $P_{E\mathbf{X}'} = P_{G\mathbf{Z}'}$: a generalization of Theorem 1. When E and G are deterministic and optimal, Theorem 2 – that E and G invert one another – can also be generalized: $\exists_{\mathbf{z} \in \hat{\Omega}_{\mathbf{Z}}} \{E(\mathbf{x}) = g_{\mathbf{Z}}(\mathbf{z}) \wedge G(\mathbf{z}) = g_{\mathbf{X}}(\mathbf{x})\}$ for $P_{\mathbf{X}}$-almost every $\mathbf{x} \in \Omega_{\mathbf{X}}$, and $\exists_{\mathbf{x} \in \hat{\Omega}_{\mathbf{X}}} \{E(\mathbf{x}) = g_{\mathbf{Z}}(\mathbf{z}) \wedge G(\mathbf{z}) = g_{\mathbf{X}}(\mathbf{x})\}$ for $P_{\mathbf{Z}}$-almost every $\mathbf{z} \in \Omega_{\mathbf{Z}}$.

4 Evaluation

We evaluate the feature learning capabilities of BiGANs by first training them unsupervised as described in Section 3.4, then transferring the encoder's learned feature representations for use in auxiliary supervised learning tasks. To demonstrate that BiGANs are able to learn meaningful feature representations both on arbitrary data vectors, where the model is agnostic to any underlying structure, as well as very high-dimensional and complex distributions, we evaluate on both permutation-invariant MNIST (LeCun et al., 1998) and on the high-resolution natural images of ImageNet (Russakovsky et al., 2015).

In all experiments, each module D, G, and E is a parametric deep (multi-layer) network. The BiGAN discriminator $D(\mathbf{x}, \mathbf{z})$ takes data $\mathbf{x}$ as its initial input, and at each linear layer thereafter, the latent representation $\mathbf{z}$ is transformed using a learned linear transformation to the hidden layer dimension and added to the non-linearity input.

4.1 Baseline methods

Besides the BiGAN framework presented above, we considered alternative approaches to learning feature representations using different GAN variants.

Discriminator The discriminator D in a standard GAN takes data samples $\mathbf{x} \sim p_{\mathbf{X}}$ as input, making its learned intermediate representations natural candidates as feature representations for related tasks.

BiGAN	D	LR	JLR	AE (ℓ_2)	AE (ℓ_1)
97.39	97.30	97.44	97.13	97.58	97.63

Table 1: One Nearest Neighbors (1NN) classification accuracy (%) on the permutation-invariant MNIST (LeCun et al., 1998) test set in the feature space learned by BiGAN, Latent Regressor (LR), Joint Latent Regressor (JLR), and an autoencoder (AE) using an ℓ_1 or ℓ_2 distance.

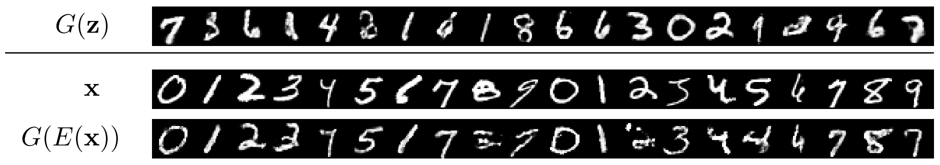

Figure 2: Qualitative results for permutation-invariant MNIST BiGAN training, including generator samples $G(\mathbf{z})$, real data $\mathbf{x}$, and corresponding reconstructions $G(E(\mathbf{x}))$.

This alternative is appealing as it requires no additional machinery, and is the approach used for unsupervised feature learning in Radford et al. (2016). On the other hand, it is not clear that the task of distinguishing between real and generated data requires or benefits from intermediate representations that are useful as semantic feature representations. In fact, if G successfully generates the true data distribution $p_{\mathbf{X}}(\mathbf{x})$, D may ignore the input data entirely and predict $P(Y = 1) = P(Y = 1|\mathbf{x}) = \frac{1}{2}$ unconditionally, not learning any meaningful intermediate representations.

Latent regressor We consider an alternative encoder training by minimizing a reconstruction loss $\mathcal{L}(\mathbf{z}, E(G(\mathbf{z})))$, after or jointly during a regular GAN training, called latent regressor or joint latent regressor respectively. We use a sigmoid cross entropy loss $\mathcal{L}$ as it naturally maps to a uniformly distributed output space. Intuitively, a drawback of this approach is that, unlike the encoder in a BiGAN, the latent regressor encoder E is trained only on generated samples $G(\mathbf{z})$, and never "sees" real data $\mathbf{x} \sim p_{\mathbf{X}}$. While this may not be an issue in the theoretical optimum where $p_G(\mathbf{x}) = p_{\mathbf{X}}(\mathbf{x})$ exactly – i.e., G perfectly generates the data distribution $p_{\mathbf{X}}$ – in practice, for highly complex data distributions $p_{\mathbf{X}}$, such as the distribution of natural images, the generator will almost never achieve this perfect result. The fact that the real data $\mathbf{x}$ are never input to this type of encoder limits its utility as a feature representation for related tasks, as shown later in this section.

4.2 Permutation-invariant MNIST

We first present results on permutation-invariant MNIST (LeCun et al., 1998). In the permutation-invariant setting, each 28×28 digit image must be treated as an unstructured 784D vector (Goodfellow et al., 2013). In our case, this condition is met by designing each module as a multi-layer perceptron (MLP), agnostic to the underlying spatial structure in the data (as opposed to a convnet, for example). See Appendix C.1 for more architectural and training details. We set the latent distribution $p_{\mathbf{Z}} = [\text{U}(-1, 1)]^{50}$ – a 50D continuous uniform distribution.

Table 1 compares the encoding learned by a BiGAN-trained encoder E with the baselines described in Section 4.1, as well as autoencoders (Hinton & Salakhutdinov, 2006) trained directly to minimize either ℓ_2 or ℓ_1 reconstruction error. The same architecture and optimization algorithm is used across all methods. All methods, including BiGAN, perform at roughly the same level. This result is not overly surprising given the relative simplicity of MNIST digits. For example, digits generated by G in a GAN nearly perfectly match the data distribution (qualitatively), making the latent regressor (LR) baseline method a reasonable choice, as argued in Section 4.1. Qualitative results are presented in Figure 2.

4.3 ImageNet

Next, we present results from training BiGANs on ImageNet LSVRC (Russakovsky et al., 2015), a large-scale database of natural images. GANs trained on ImageNet cannot perfectly reconstruct

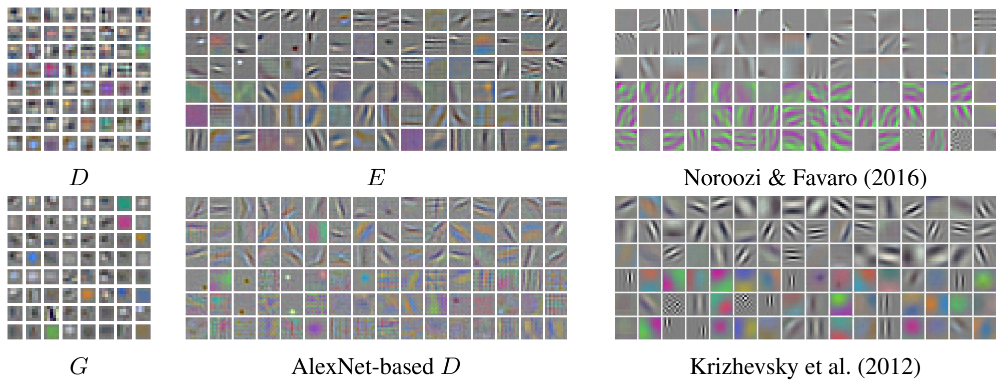

Figure 3: The convolutional filters learned by the three modules (D, G, and E) of a BiGAN (left, top-middle) trained on the ImageNet (Russakovsky et al., 2015) database. We compare with the filters learned by a discriminator D trained with the same architecture (bottom-middle), as well as the filters reported by Noroozi & Favaro (2016), and by Krizhevsky et al. (2012) for fully supervised ImageNet training (right).

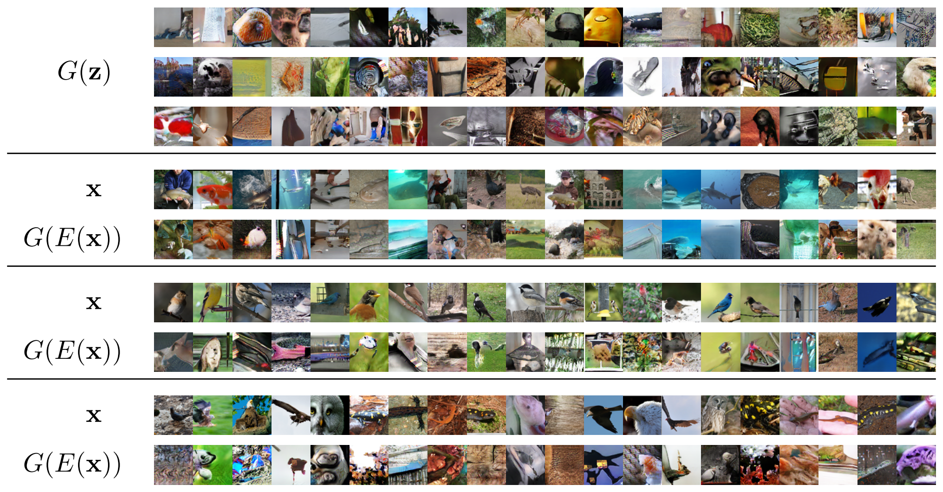

Figure 4: Qualitative results for ImageNet BiGAN training, including generator samples $G(\mathbf{z})$, real data $\mathbf{x}$, and corresponding reconstructions $G(E(\mathbf{x}))$.

the data, but often capture some interesting aspects. Here, each of D, G, and E is a convnet. In all experiments, the encoder E architecture follows AlexNet (Krizhevsky et al., 2012) through the fifth and last convolution layer (*conv5*). We also experiment with an AlexNet-based discriminator D as a baseline feature learning approach. We set the latent distribution $p_{\mathbf{Z}} = [\mathrm{U}(-1,1)]^{200}$ – a 200D continuous uniform distribution. Additionally, we experiment with higher resolution encoder input images – 112×112 rather than the 64×64 used elsewhere – using the generalization described in Section 3.5. See Appendix C.2 for more architectural and training details.

Qualitative results The convolutional filters learned by each of the three modules are shown in Figure 3. We see that the filters learned by the encoder E have clear Gabor-like structure, similar to those originally reported for the fully supervised AlexNet model (Krizhevsky et al., 2012). The filters also have similar "grouping" structure where one half (the bottom half, in this case) is more color sensitive, and the other half is more edge sensitive. (This separation of the filters occurs due to the AlexNet architecture maintaining two separate filter paths for computational efficiency.)

In Figure 4 we present sample generations $G(\mathbf{z})$, as well as real data samples $\mathbf{x}$ and their BiGAN reconstructions $G(E(\mathbf{x}))$. The reconstructions, while certainly imperfect, demonstrate empirically that

	conv1	conv2	conv3	conv4	conv5
Random (Noroozi & Favaro, 2016)	48.5	41.0	34.8	27.1	12.0
Wang & Gupta (2015)	51.8	46.9	42.8	38.8	29.8
Doersch et al. (2015)	53.1	47.6	48.7	**45.6**	30.4
Noroozi & Favaro (2016)*	57.1	56.0	52.4	48.3	38.1
BiGAN (ours)	**56.2**	**54.4**	**49.4**	43.9	33.3
BiGAN, 112×112 E (ours)	55.3	53.2	49.3	44.4	**34.8**

Table 2: Classification accuracy (%) for the ImageNet LSVRC (Russakovsky et al., 2015) validation set with various portions of the network frozen, or reinitialized and trained from scratch, following the evaluation from Noroozi & Favaro (2016). In, e.g., the *conv3* column, the first three layers – conv1 through conv3 – are transferred and frozen, and the last layers – conv4, conv5, and fully connected layers – are reinitialized and trained fully supervised for ImageNet classification. BiGAN is competitive with these contemporary visual feature learning methods, despite its generality. (*Results from Noroozi & Favaro (2016) are not directly comparable to those of the other methods as a different base convnet architecture with larger intermediate feature maps is used.)

the BiGAN encoder E and generator G learn approximate inverse mappings, as shown theoretically in Theorem 2. In Appendix C.2, we present nearest neighbors in the BiGAN learned feature space.

ImageNet classification Following Noroozi & Favaro (2016), we evaluate by freezing the first N layers of our pretrained network and randomly reinitializing and training the remainder fully supervised for ImageNet classification. Results are reported in Table 2.

VOC classification, detection, and segmentation We evaluate the transferability of BiGAN representations to the PASCAL VOC (Everingham et al., 2014) computer vision benchmark tasks, including classification, object detection, and semantic segmentation. The classification task involves simple binary prediction of presence or absence in a given image for each of 20 object categories. The object detection and semantic segmentation tasks go a step further by requiring the objects to be localized, with semantic segmentation requiring this at the finest scale: pixelwise prediction of object identity. For detection, the pretrained model is used as the initialization for *Fast R-CNN* (Girshick, 2015) (FRCN) training; and for semantic segmentation, the model is used as the initialization for *Fully Convolutional Network* (Long et al., 2015) (FCN) training, in each case replacing the *AlexNet* (Krizhevsky et al., 2012) model trained fully supervised for ImageNet classification. We report results on each of these tasks in Table 3, comparing BiGANs with contemporary approaches to unsupervised (Krähenbühl et al., 2016) and self-supervised (Doersch et al., 2015; Agrawal et al., 2015; Wang & Gupta, 2015; Pathak et al., 2016) feature learning in the visual domain, as well as the baselines discussed in Section 4.1.

4.4 Discussion

Despite making no assumptions about the underlying structure of the data, the BiGAN unsupervised feature learning framework offers a representation competitive with existing self-supervised and even weakly supervised feature learning approaches for visual feature learning, while still being a purely generative model with the ability to sample data $\mathbf{x}$ and predict latent representation $\mathbf{z}$. Furthermore, BiGANs outperform the discriminator (D) and latent regressor (LR) baselines discussed in Section 4.1, confirming our intuition that these approaches may not perform well in the regime of highly complex data distributions such as that of natural images. The version in which the encoder takes a higher resolution image than output by the generator (*BiGAN* 112×112 E) performs better still, and this strategy is not possible under the LR and D baselines as each of those modules take generator outputs as their input.

Although existing self-supervised approaches have shown impressive performance and thus far tended to outshine purely unsupervised approaches in the complex domain of high-resolution images, purely unsupervised approaches to feature learning or pre-training have several potential benefits.

		Classification (% mAP)			*FRCN* Detection (% mAP)	*FCN* Segmentation (% mIU)
	trained layers	fc8	fc6-8	all	all	all
sup.	ImageNet (Krizhevsky et al., 2012)	**77.0**	**78.8**	**78.3**	**56.8**	**48.0**
self-sup.	Agrawal et al. (2015)	31.2	31.0	54.2	43.9	-
	Pathak et al. (2016)	30.5	34.6	56.5	44.5	**30.0**
	Wang & Gupta (2015)	28.4	**55.6**	63.1	47.4	-
	Doersch et al. (2015)	**44.7**	55.1	**65.3**	**51.1**	-
unsup.	k-means (Krähenbühl et al., 2016)	32.0	39.2	56.6	45.6	32.6
	Discriminator (D)	30.7	40.5	56.4	-	-
	Latent Regressor (LR)	36.9	47.9	57.1	-	-
	Joint LR	37.1	47.9	56.5	-	-
	Autoencoder (ℓ_2)	24.8	16.0	53.8	41.9	-
	BiGAN (ours)	37.5	48.7	58.9	46.2	34.9
	BiGAN, 112×112 E (ours)	**40.7**	**52.3**	**60.1**	**46.9**	**35.2**

Table 3: Classification and Fast R-CNN (Girshick, 2015) detection results for the PASCAL VOC 2007 (Everingham et al., 2014) test set, and FCN (Long et al., 2015) segmentation results on the PASCAL VOC 2012 validation set, under the standard mean average precision (mAP) or mean intersection over union (mIU) metrics for each task. Classification models are trained with various portions of the *AlexNet* (Krizhevsky et al., 2012) model frozen. In the *fc8* column, only the linear classifier (a multinomial logistic regression) is learned – in the case of BiGAN, on top of randomly initialized fully connected (FC) layers *fc6* and *fc7*. In the *fc6-8* column, all three FC layers are trained fully supervised with all convolution layers frozen. Finally, in the *all* column, the entire network is "fine-tuned". BiGAN outperforms other unsupervised (*unsup.*) feature learning approaches, including the GAN-based baselines described in Section 4.1, and despite its generality, is competitive with contemporary self-supervised (*self-sup.*) feature learning approaches specific to the visual domain.

BiGAN and other unsupervised learning approaches are agnostic to the domain of the data. The self-supervised approaches are specific to the visual domain, in some cases requiring weak supervision from video unavailable in images alone. For example, the methods are not applicable in the permutation-invariant MNIST setting explored in Section 4.2, as the data are treated as flat vectors rather than 2D images.

Furthermore, BiGAN and other unsupervised approaches needn't suffer from domain shift between the pre-training task and the transfer task, unlike self-supervised methods in which some aspect of the data is normally removed or corrupted in order to create a non-trivial prediction task. In the context prediction task (Doersch et al., 2015), the network sees only small image patches – the global image structure is unobserved. In the context encoder or inpainting task (Pathak et al., 2016), each image is corrupted by removing large areas to be filled in by the prediction network, creating inputs with dramatically different appearance from the uncorrupted natural images seen in the transfer tasks.

Other approaches (Agrawal et al., 2015; Wang & Gupta, 2015) rely on auxiliary information unavailable in the static image domain, such as video, egomotion, or tracking. Unlike BiGAN, such approaches cannot learn feature representations from unlabeled static images.

We finally note that the results presented here constitute only a preliminary exploration of the space of model architectures possible under the BiGAN framework, and we expect results to improve significantly with advancements in generative image models and discriminative convolutional networks alike.

Acknowledgments

The authors thank Evan Shelhamer, Jonathan Long, and other Berkeley Vision labmates for helpful discussions throughout this work. This work was supported by DARPA, AFRL, DoD MURI award N000141110688, NSF awards IIS-1427425 and IIS-1212798, and the Berkeley Artificial Intelligence Research laboratory. The GPUs used for this work were donated by NVIDIA.

REFERENCES

Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *ICCV*, 2015.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

Emily L. Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a Laplacian pyramid of adversarial networks. In *NIPS*, 2015.

Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. *arXiv:1606.00704*, 2016.

Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes challenge: A retrospective. *IJCV*, 2014.

Ross Girshick. Fast R-CNN. In *ICCV*, 2015.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *ICML*, 2013.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. In *ICASSP*, 2013.

Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006.

Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 2006.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Philipp Krähenbühl, Carl Doersch, Jeff Donahue, and Trevor Darrell. Data-dependent initializations of convolutional neural networks. In *ICLR*, 2016.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 1998.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

Ali Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *CVPR Workshops*, 2014.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Fei-Fei Li. ImageNet large scale visual recognition challenge. *IJCV*, 2015.

Ruslan Salakhutdinov and Geoffrey E. Hinton. Deep Boltzmann machines. In *AISTATS*, 2009.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv:1605.02688*, 2016.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey E. Hinton. Grammar as a foreign language. In *NIPS*, 2015.

Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

APPENDIX A ADDITIONAL PROOFS

A.1 PROOF OF PROPOSITION 1 (OPTIMAL DISCRIMINATOR)

Proposition 1 *For any E and G, the optimal discriminator $D^*_{EG} := \arg\max_D V(D, E, G)$ is the Radon-Nikodym derivative $f_{EG} := \frac{\mathrm{d}P_{E\mathbf{X}}}{\mathrm{d}(P_{E\mathbf{X}}+P_{G\mathbf{Z}})} : \Omega \mapsto [0, 1]$ of measure $P_{E\mathbf{X}}$ with respect to measure $P_{E\mathbf{X}} + P_{G\mathbf{Z}}$.*

Proof. For measures P and Q on space Ω, with P absolutely continuous with respect to Q, the RN derivative $f_{PQ} := \frac{\mathrm{d}P}{\mathrm{d}Q}$ exists, and we have

$$\mathbb{E}_{\mathbf{x}\sim P}[g(\mathbf{x})] = \int_\Omega g\,\mathrm{d}P = \int_\Omega g\frac{\mathrm{d}P}{\mathrm{d}Q}\,\mathrm{d}Q = \int_\Omega g f_{PQ}\,\mathrm{d}Q = \mathbb{E}_{\mathbf{x}\sim Q}[f_{PQ}(\mathbf{x})g(\mathbf{x})]. \tag{4}$$

Let the probability measure $P_{EG} := \frac{P_{E\mathbf{X}}+P_{G\mathbf{Z}}}{2}$ denote the average of measures $P_{E\mathbf{X}}$ and $P_{G\mathbf{Z}}$. Both $P_{E\mathbf{X}}$ and $P_{G\mathbf{Z}}$ are each absolutely continuous with respect to P_{EG}. Hence the RN derivatives $f_{EG} := \frac{\mathrm{d}P_{E\mathbf{X}}}{\mathrm{d}(P_{E\mathbf{X}}+P_{G\mathbf{Z}})} = \frac{1}{2}\frac{\mathrm{d}P_{E\mathbf{X}}}{\mathrm{d}P_{EG}}$ and $f_{GE} := \frac{\mathrm{d}P_{G\mathbf{Z}}}{\mathrm{d}(P_{E\mathbf{X}}+P_{G\mathbf{Z}})} = \frac{1}{2}\frac{\mathrm{d}P_{G\mathbf{Z}}}{\mathrm{d}P_{EG}}$ exist and sum to 1:

$$f_{EG} + f_{GE} = \frac{\mathrm{d}P_{E\mathbf{X}}}{\mathrm{d}(P_{E\mathbf{X}}+P_{G\mathbf{Z}})} + \frac{\mathrm{d}P_{G\mathbf{Z}}}{\mathrm{d}(P_{E\mathbf{X}}+P_{G\mathbf{Z}})} = \frac{\mathrm{d}(P_{E\mathbf{X}}+P_{G\mathbf{Z}})}{\mathrm{d}(P_{E\mathbf{X}}+P_{G\mathbf{Z}})} = 1. \tag{5}$$

We use (4) and (5) to rewrite the objective V (3) as a single expectation under measure P_{EG}:

$$\begin{aligned}
V(D, E, G) &= \mathbb{E}_{(\mathbf{x},\mathbf{z})\sim P_{E\mathbf{X}}}[\log D(\mathbf{x},\mathbf{z})] + \mathbb{E}_{(\mathbf{x},\mathbf{z})\sim P_{G\mathbf{Z}}}[\log(1 - D(\mathbf{x},\mathbf{z}))] \\
&= \mathbb{E}_{(\mathbf{x},\mathbf{z})\sim P_{EG}}[\underbrace{2f_{EG}}_{\frac{\mathrm{d}P_{E\mathbf{X}}}{\mathrm{d}P_{EG}}}(\mathbf{x},\mathbf{z})\log D(\mathbf{x},\mathbf{z})] + \mathbb{E}_{(\mathbf{x},\mathbf{z})\sim P_{EG}}[\underbrace{2f_{GE}}_{\frac{\mathrm{d}P_{G\mathbf{Z}}}{\mathrm{d}P_{EG}}}(\mathbf{x},\mathbf{z})\log(1 - D(\mathbf{x},\mathbf{z}))] \\
&= 2\,\mathbb{E}_{(\mathbf{x},\mathbf{z})\sim P_{EG}}[f_{EG}(\mathbf{x},\mathbf{z})\log D(\mathbf{x},\mathbf{z}) + f_{GE}(\mathbf{x},\mathbf{z})\log(1 - D(\mathbf{x},\mathbf{z}))] \\
&= 2\,\mathbb{E}_{(\mathbf{x},\mathbf{z})\sim P_{EG}}[f_{EG}(\mathbf{x},\mathbf{z})\log D(\mathbf{x},\mathbf{z}) + (1 - f_{EG}(\mathbf{x},\mathbf{z}))\log(1 - D(\mathbf{x},\mathbf{z}))].
\end{aligned}$$

Note that $\arg\max_y\{a\log y + (1-a)\log(1-y)\} = a$ for any $a \in [0, 1]$. Thus, $D^*_{EG} = f_{EG}$. □

A.2 PROOF OF PROPOSITION 2 (ENCODER AND GENERATOR OBJECTIVE)

Proposition 2 *The encoder and generator's objective for an optimal discriminator $C(E, G) := \max_D V(D, E, G) = V(D^*_{EG}, E, G)$ can be rewritten in terms of the Jensen-Shannon divergence between measures $P_{E\mathbf{X}}$ and $P_{G\mathbf{Z}}$ as $C(E, G) = 2\,\mathrm{D_{JS}}(P_{E\mathbf{X}}\,||\,P_{G\mathbf{Z}}) - \log 4$.*

Proof. Using Proposition 1 along with (5) ($1 - D^*_{EG} = 1 - f_{EG} = f_{GE}$) we rewrite the objective

$$\begin{aligned}
C(E, G) &= \max\nolimits_D V(D, E, G) = V(D^*_{EG}, E, G) \\
&= \mathbb{E}_{(\mathbf{x},\mathbf{z})\sim P_{E\mathbf{X}}}[\log D^*_{EG}(\mathbf{x},\mathbf{z})] + \mathbb{E}_{(\mathbf{x},\mathbf{z})\sim P_{G\mathbf{Z}}}[\log(1 - D^*_{EG}(\mathbf{x},\mathbf{z}))] \\
&= \mathbb{E}_{(\mathbf{x},\mathbf{z})\sim P_{E\mathbf{X}}}[\log f_{EG}(\mathbf{x},\mathbf{z})] + \mathbb{E}_{(\mathbf{x},\mathbf{z})\sim P_{G\mathbf{Z}}}[\log f_{GE}(\mathbf{x},\mathbf{z})] \\
&= \mathbb{E}_{(\mathbf{x},\mathbf{z})\sim P_{E\mathbf{X}}}[\log(2f_{EG}(\mathbf{x},\mathbf{z}))] + \mathbb{E}_{(\mathbf{x},\mathbf{z})\sim P_{G\mathbf{Z}}}[\log(2f_{GE}(\mathbf{x},\mathbf{z}))] - \log 4 \\
&= \mathrm{D_{KL}}(P_{E\mathbf{X}}\,||\,P_{EG}) + \mathrm{D_{KL}}(P_{G\mathbf{Z}}\,||\,P_{EG}) - \log 4 \\
&= \mathrm{D_{KL}}\left(P_{E\mathbf{X}}\,||\,\tfrac{P_{E\mathbf{X}}+P_{G\mathbf{Z}}}{2}\right) + \mathrm{D_{KL}}\left(P_{G\mathbf{Z}}\,||\,\tfrac{P_{E\mathbf{X}}+P_{G\mathbf{Z}}}{2}\right) - \log 4 \\
&= 2\,\mathrm{D_{JS}}(P_{E\mathbf{X}}\,||\,P_{G\mathbf{Z}}) - \log 4. \;\square
\end{aligned}$$

A.3 MEASURE DEFINITIONS FOR DETERMINISTIC E AND G

While Theorem 1 and Propositions 1 and 2 hold for any encoder $p_E(\mathbf{z}|\mathbf{x})$ and generator $p_G(\mathbf{x}|\mathbf{z})$, stochastic or deterministic, Theorems 2 and 3 assume the encoder E and generator G are deterministic functions; i.e., with conditionals $p_E(\mathbf{z}|\mathbf{x}) = \delta(\mathbf{z} - E(\mathbf{x}))$ and $p_G(\mathbf{x}|\mathbf{z}) = \delta(\mathbf{x} - G(\mathbf{z}))$ defined as δ functions.

For use in the proofs of those theorems, we simplify the definitions of measures $P_{E\mathbf{X}}$ and $P_{G\mathbf{Z}}$ given in Section 3 for the case of deterministic functions E and G below:

$$\begin{aligned}
P_{E\mathbf{X}}(R) &= \int_{\Omega_\mathbf{X}} p_\mathbf{X}(\mathbf{x}) \int_{\Omega_\mathbf{Z}} p_E(\mathbf{z}|\mathbf{x}) \mathbf{1}_{[(\mathbf{x},\mathbf{z})\in R]} \,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{x} \\
&= \int_{\Omega_\mathbf{X}} p_\mathbf{X}(\mathbf{x}) \left(\int_{\Omega_\mathbf{Z}} \delta(\mathbf{z} - E(\mathbf{x})) \mathbf{1}_{[(\mathbf{x},\mathbf{z})\in R]} \,\mathrm{d}\mathbf{z} \right) \mathrm{d}\mathbf{x} \\
&= \int_{\Omega_\mathbf{X}} p_\mathbf{X}(\mathbf{x}) \mathbf{1}_{[(\mathbf{x},E(\mathbf{x}))\in R]} \,\mathrm{d}\mathbf{x} \\
P_{G\mathbf{Z}}(R) &= \int_{\Omega_\mathbf{Z}} p_\mathbf{Z}(\mathbf{z}) \int_{\Omega_\mathbf{X}} p_G(\mathbf{x}|\mathbf{z}) \mathbf{1}_{[(\mathbf{x},\mathbf{z})\in R]} \,\mathrm{d}\mathbf{x}\,\mathrm{d}\mathbf{z} \\
&= \int_{\Omega_\mathbf{Z}} p_\mathbf{Z}(\mathbf{z}) \left(\int_{\Omega_\mathbf{X}} \delta(\mathbf{x} - G(\mathbf{z})) \mathbf{1}_{[(\mathbf{x},\mathbf{z})\in R]} \,\mathrm{d}\mathbf{x} \right) \mathrm{d}\mathbf{z} \\
&= \int_{\Omega_\mathbf{Z}} p_\mathbf{Z}(\mathbf{z}) \mathbf{1}_{[(G(\mathbf{z}),\mathbf{z})\in R]} \,\mathrm{d}\mathbf{z}
\end{aligned}$$

A.4 Proof of Theorem 2 (optimal generator and encoder are inverses)

Theorem 2 *If E and G are an optimal encoder and generator, then $E = G^{-1}$ almost everywhere; that is, $G(E(\mathbf{x})) = \mathbf{x}$ for $P_\mathbf{X}$-almost every $\mathbf{x} \in \Omega_\mathbf{X}$, and $E(G(\mathbf{z})) = \mathbf{z}$ for $P_\mathbf{Z}$-almost every $\mathbf{z} \in \Omega_\mathbf{Z}$.*

Proof. Let $R^0_\mathbf{X} := \{\mathbf{x} \in \Omega_\mathbf{X} : \mathbf{x} \neq G(E(\mathbf{x}))\}$ be the region of $\Omega_\mathbf{X}$ in which the inversion property $\mathbf{x} = G(E(\mathbf{x}))$ does *not* hold. We will show that, for optimal E and G, $R^0_\mathbf{X}$ has measure zero under $P_\mathbf{X}$ (i.e., $P_\mathbf{X}(R^0_\mathbf{X}) = 0$) and therefore $\mathbf{x} = G(E(\mathbf{x}))$ holds $P_\mathbf{X}$-almost everywhere.

Let $R^0 := \{(\mathbf{x},\mathbf{z}) \in \Omega : \mathbf{z} = E(\mathbf{x}) \wedge \mathbf{x} \in R^0_\mathbf{X}\}$ be the region of Ω such that $(\mathbf{x}, E(\mathbf{x})) \in R^0$ if and only if $\mathbf{x} \in R^0_\mathbf{X}$. We'll use the definitions of $P_{E\mathbf{X}}$ and $P_{G\mathbf{Z}}$ for deterministic E and G (Appendix A.3), and the fact that $P_{E\mathbf{X}} = P_{G\mathbf{Z}}$ for optimal E and G (Theorem 1).

$$\begin{aligned}
P_\mathbf{X}(R^0_\mathbf{X}) &= \int_{\Omega_\mathbf{X}} p_\mathbf{X}(\mathbf{x}) \mathbf{1}_{\left[\mathbf{x}\in R^0_\mathbf{X}\right]} \,\mathrm{d}\mathbf{x} \\
&= \int_{\Omega_\mathbf{X}} p_\mathbf{X}(\mathbf{x}) \mathbf{1}_{[(\mathbf{x},E(\mathbf{x}))\in R^0]} \,\mathrm{d}\mathbf{x} \\
&= P_{E\mathbf{X}}(R^0) \\
&= P_{G\mathbf{Z}}(R^0) \\
&= \int_{\Omega_\mathbf{Z}} p_\mathbf{Z}(\mathbf{z}) \mathbf{1}_{[(G(\mathbf{z}),\mathbf{z})\in R^0]} \,\mathrm{d}\mathbf{z} \\
&= \int_{\Omega_\mathbf{Z}} p_\mathbf{Z}(\mathbf{z}) \mathbf{1}_{\left[\mathbf{z}=E(G(\mathbf{z}))\, \wedge\, G(\mathbf{z})\in R^0_\mathbf{X}\right]} \,\mathrm{d}\mathbf{z} \\
&= \int_{\Omega_\mathbf{Z}} p_\mathbf{Z}(\mathbf{z}) \underbrace{\mathbf{1}_{[\mathbf{z}=E(G(\mathbf{z}))\, \wedge\, G(\mathbf{z})\neq G(E(G(\mathbf{z})))]}}_{=0 \text{ for any } \mathbf{z}, \text{ as } \mathbf{z}=E(G(\mathbf{z})) \implies G(\mathbf{z})=G(E(G(\mathbf{z})))} \,\mathrm{d}\mathbf{z} \\
&= 0.
\end{aligned}$$

Hence region $R^0_\mathbf{X}$ has measure zero ($P_\mathbf{X}(R^0_\mathbf{X}) = 0$), and the inversion property $\mathbf{x} = G(E(\mathbf{x}))$ holds $P_\mathbf{X}$-almost everywhere.

An analogous argument shows that $R^0_\mathbf{Z} := \{\mathbf{z} \in \Omega_\mathbf{Z} : \mathbf{z} \neq E(G(\mathbf{z}))\}$ has measure zero on $P_\mathbf{Z}$ (i.e., $P_\mathbf{Z}(R^0_\mathbf{Z}) = 0$) and therefore $\mathbf{z} = E(G(\mathbf{z}))$ holds $P_\mathbf{Z}$-almost everywhere. □

A.5 Proof of Theorem 3 (relationship to autoencoders)

As shown in Proposition 2 (Section 3), the BiGAN objective is equivalent to the Jensen-Shannon divergence between $P_{E\mathbf{X}}$ and $P_{G\mathbf{Z}}$. We now go a step further and show that this Jensen-Shannon divergence is closely related to a standard autoencoder loss. Omitting the $\frac{1}{2}$ scale factor, a KL divergence term of the Jensen-Shannon divergence is given as

$$\begin{aligned}
\mathrm{D_{KL}}\left(P_{E\mathbf{X}} \,||\, \tfrac{P_{E\mathbf{X}}+P_{G\mathbf{Z}}}{2} \right) &= \log 2 + \int_\Omega \log \frac{\mathrm{d}P_{E\mathbf{X}}}{\mathrm{d}(P_{E\mathbf{X}} + P_{G\mathbf{Z}})} \,\mathrm{d}P_{E\mathbf{X}} \\
&= \log 2 + \int_\Omega \log f \,\mathrm{d}P_{E\mathbf{X}}, \qquad (6)
\end{aligned}$$

where we abbreviate as f the Radon-Nikodym derivative $f_{EG} := \frac{\mathrm{d}P_{E\mathbf{X}}}{\mathrm{d}(P_{E\mathbf{X}}+P_{G\mathbf{Z}})} \in [0,1]$ defined in Proposition 1 for most of this proof.

We'll make use of the definitions of $P_{E\mathbf{X}}$ and $P_{G\mathbf{Z}}$ for deterministic E and G found in Appendix A.3. The integral term of the KL divergence expression given in (6) over a particular region $R \subseteq \Omega$ will be denoted by

$$F(R) := \int_R \log \frac{\mathrm{d}P_{E\mathbf{X}}}{\mathrm{d}\left(P_{E\mathbf{X}} + P_{G\mathbf{Z}}\right)} \, \mathrm{d}P_{E\mathbf{X}} = \int_R \log f \, \mathrm{d}P_{E\mathbf{X}}.$$

Next we will show that $f > 0$ holds $P_{E\mathbf{X}}$-almost everywhere, and hence F is always well defined and finite. We then show that F is equivalent to an autoencoder-like reconstruction loss function.

Proposition 3 *$f > 0$ $P_{E\mathbf{X}}$-almost everywhere.*

Proof. Let $R^{f=0} := \{(\mathbf{x}, \mathbf{z}) \in \Omega : f(\mathbf{x}, \mathbf{z}) = 0\}$ be the region of Ω in which $f = 0$. Using the definition of the Radon-Nikodym derivative f, the measure $P_{E\mathbf{X}}(R^{f=0}) = \int_{R^{f=0}} f \, \mathrm{d}(P_{E\mathbf{X}} + P_{G\mathbf{Z}}) = \int_{R^{f=0}} 0 \, \mathrm{d}(P_{E\mathbf{X}} + P_{G\mathbf{Z}}) = 0$ is zero. Hence $f > 0$ $P_{E\mathbf{X}}$-almost everywhere. $\square$

Proposition 3 ensures that $\log f$ is defined $P_{E\mathbf{X}}$-almost everywhere, and $F(R)$ is well-defined. Next we will show that $F(R)$ mimics an autoencoder with ℓ_0 loss, meaning F is zero for any region in which $G(E(\mathbf{x})) \neq \mathbf{x}$, and non-zero otherwise.

Proposition 4 *The KL divergence F outside the support of $P_{G\mathbf{Z}}$ is zero: $F(\Omega \setminus \operatorname{supp}(P_{G\mathbf{Z}})) = 0$.*

We'll first show that in region $R_S := \Omega \setminus \operatorname{supp}(P_{G\mathbf{Z}})$, we have $f = 1$ $P_{E\mathbf{X}}$-almost everywhere. Let $R^{f<1} := \{(\mathbf{x}, \mathbf{z}) \in R_S : f(\mathbf{x}, \mathbf{z}) < 1\}$ be the region of R_S in which $f < 1$. Let's assume that $P_{E\mathbf{X}}(R^{f<1}) > 0$ has non-zero measure. Then, using the definition of the Radon-Nikodym derivative,

$$\begin{aligned} P_{E\mathbf{X}}(R^{f<1}) = \textstyle\int_{R^{f<1}} f \, \mathrm{d}(P_{E\mathbf{X}} + P_{G\mathbf{Z}}) = \int_{R^{f<1}} \underbrace{f}_{\leq \varepsilon < 1} \, \mathrm{d}P_{E\mathbf{X}} + \underbrace{\textstyle\int_{R^{f<1}} f \, \mathrm{d}P_{G\mathbf{Z}}}_{0} \leq \varepsilon P_{E\mathbf{X}}(R^{f<1}) \\ < P_{E\mathbf{X}}(R^{f<1}), \end{aligned}$$

where ε is a constant smaller than 1. But $P_{E\mathbf{X}}(R^{f<1}) < P_{E\mathbf{X}}(R^{f<1})$ is a contradiction; hence $P_{E\mathbf{X}}(R^{f<1}) = 0$ and $f = 1$ $P_{E\mathbf{X}}$-almost everywhere in R_S, implying $\log f = 0$ $P_{E\mathbf{X}}$-almost everywhere in R_S. Hence $F(R_S) = 0$. $\square$

By definition, $F(\Omega \setminus \operatorname{supp}(P_{E\mathbf{X}})) = 0$ is also zero. The only region where F might be non-zero is $R^1 := \operatorname{supp}(P_{E\mathbf{X}}) \cap \operatorname{supp}(P_{G\mathbf{Z}})$.

Proposition 5 *$f < 1$ $P_{E\mathbf{X}}$-almost everywhere in R^1.*

Let $R^{f=1} := \left\{(\mathbf{x}, \mathbf{z}) \in R^1 : f(\mathbf{x}, \mathbf{z}) = 1\right\}$ be the region in which $f = 1$. Let's assume the set $R^{f=1} \neq \emptyset$ is not empty. By definition of the support[1], $P_{E\mathbf{X}}(R^{f=1}) > 0$ and $P_{G\mathbf{Z}}(R^{f=1}) > 0$. The Radon-Nikodym derivative on $R^{f=1}$ is then given by

$$P_{E\mathbf{X}}(R^{f=1}) = \textstyle\int_{R^{f=1}} f \, \mathrm{d}(P_{E\mathbf{X}} + P_{G\mathbf{Z}}) = \int_{R^{f=1}} 1 \, \mathrm{d}(P_{E\mathbf{X}} + P_{G\mathbf{Z}}) = P_{E\mathbf{X}}(R^{f=1}) + P_{G\mathbf{Z}}(R^{f=1}),$$

which implies $P_{G\mathbf{Z}}(R^{f=1}) = 0$ and contradicts the definition of support. Hence $R^{f=1} = \emptyset$ and $f < 1$ $P_{E\mathbf{X}}$-almost everywhere on R^1, implying $\log f < 0$ $P_{E\mathbf{X}}$-almost everywhere. $\square$

Theorem 3 *The encoder and generator objective given an optimal discriminator $C(E, G) := \max_D V(D, E, G)$ can be rewritten as an ℓ_0 autoencoder loss function*

$$\begin{aligned} C(E, G) = \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{X}}} \left[\mathbf{1}_{\left[E(\mathbf{x}) \in \hat{\Omega}_{\mathbf{Z}} \wedge G(E(\mathbf{x})) = \mathbf{x}\right]} \log f_{EG}(\mathbf{x}, E(\mathbf{x})) \right] + \\ \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}} \left[\mathbf{1}_{\left[G(\mathbf{z}) \in \hat{\Omega}_{\mathbf{X}} \wedge E(G(\mathbf{z})) = \mathbf{z}\right]} \log \left(1 - f_{EG}(G(\mathbf{z}), \mathbf{z})\right) \right] \end{aligned}$$

with $\log f_{EG} \in (-\infty, 0)$ *and* $\log\left(1 - f_{EG}\right) \in (-\infty, 0)$ *$P_{E\mathbf{X}}$-almost and $P_{G\mathbf{Z}}$-almost everywhere.*

Proof. Proposition 4 ($F(\Omega \setminus \operatorname{supp}(P_{G\mathbf{Z}})) = 0$) and $F(\Omega \setminus \operatorname{supp}(P_{E\mathbf{X}})) = 0$ imply that $R^1 := \operatorname{supp}(P_{E\mathbf{X}}) \cap \operatorname{supp}(P_{G\mathbf{Z}})$ is the only region of Ω where F may be non-zero; hence $F(\Omega) = F(R^1)$.

[1]We use the definition $U \cap C \neq \emptyset \implies \mu(U \cap C) > 0$ here.

Note that

$$\begin{aligned} \operatorname{supp}(P_{E\mathbf{X}}) &= \{(\mathbf{x}, E(\mathbf{x})) : \mathbf{x} \in \hat{\Omega}_{\mathbf{X}}\} \\ \operatorname{supp}(P_{G\mathbf{Z}}) &= \{(G(\mathbf{z}), \mathbf{z}) : \mathbf{z} \in \hat{\Omega}_{\mathbf{Z}}\} \\ \implies R^1 := \operatorname{supp}(P_{E\mathbf{X}}) \cap \operatorname{supp}(P_{G\mathbf{Z}}) &= \{(\mathbf{x}, \mathbf{z}) : E(\mathbf{x}) = \mathbf{z} \wedge \mathbf{x} \in \hat{\Omega}_{\mathbf{X}} \wedge G(\mathbf{z}) = \mathbf{x} \wedge \mathbf{z} \in \hat{\Omega}_{\mathbf{Z}}\} \end{aligned}$$

So a point $(\mathbf{x}, E(\mathbf{x}))$ is in R^1 if $\mathbf{x} \in \hat{\Omega}_{\mathbf{X}}$, $E(\mathbf{x}) \in \hat{\Omega}_{\mathbf{Z}}$, and $G(E(\mathbf{x})) = \mathbf{x}$. (We can omit the $\mathbf{x} \in \hat{\Omega}_{\mathbf{X}}$ condition from inside an expectation over $P_{\mathbf{X}}$, as $P_{\mathbf{X}}$-almost all $\mathbf{x} \notin \hat{\Omega}_{\mathbf{X}}$ have 0 probability.) Therefore,

$$\begin{aligned} \mathrm{D_{KL}}\left(P_{E\mathbf{X}} \,\middle|\middle|\, \tfrac{P_{E\mathbf{X}}+P_{G\mathbf{Z}}}{2}\right) - \log 2 &= F(\Omega) = F(R^1) \\ &= \textstyle\int_{R^1} \log f(\mathbf{x}, \mathbf{z}) \, \mathrm{d}P_{E\mathbf{X}} \\ &= \textstyle\int_{\Omega} \mathbf{1}_{[(\mathbf{x},\mathbf{z}) \in R^1]} \log f(\mathbf{x}, \mathbf{z}) \, \mathrm{d}P_{E\mathbf{X}} \\ &= \mathbb{E}_{(\mathbf{x},\mathbf{z}) \sim P_{E\mathbf{X}}} \left[\mathbf{1}_{[(\mathbf{x},\mathbf{z}) \in R^1]} \log f(\mathbf{x}, \mathbf{z})\right] \\ &= \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{X}}} \left[\mathbf{1}_{[(\mathbf{x},E(\mathbf{x})) \in R^1]} \log f(\mathbf{x}, E(\mathbf{x}))\right] \\ &= \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{X}}} \left[\mathbf{1}_{\left[E(\mathbf{x}) \in \hat{\Omega}_{\mathbf{Z}} \wedge G(E(\mathbf{x}))=\mathbf{x}\right]} \log f(\mathbf{x}, E(\mathbf{x}))\right]. \end{aligned}$$

Finally, with Propositions 3 and 5, we have $f \in (0, 1)$ $P_{E\mathbf{X}}$-almost everywhere in R^1, and therefore $\log f \in (-\infty, 0)$, taking a finite and strictly negative value $P_{E\mathbf{X}}$-almost everywhere.

An analogous argument (along with the fact that $f_{EG} + f_{GE} = 1$) lets us rewrite the other KL divergence term

$$\begin{aligned} \mathrm{D_{KL}}\left(P_{G\mathbf{Z}} \,\middle|\middle|\, \tfrac{P_{E\mathbf{X}}+P_{G\mathbf{Z}}}{2}\right) - \log 2 &= \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}} \left[\mathbf{1}_{\left[G(\mathbf{z}) \in \hat{\Omega}_{\mathbf{X}} \wedge E(G(\mathbf{z}))=\mathbf{z}\right]} \log f_{GE}(G(\mathbf{z}), \mathbf{z})\right] \\ &= \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}} \left[\mathbf{1}_{\left[G(\mathbf{z}) \in \hat{\Omega}_{\mathbf{X}} \wedge E(G(\mathbf{z}))=\mathbf{z}\right]} \log\left(1 - f_{EG}(G(\mathbf{z}), \mathbf{z})\right)\right] \end{aligned}$$

The Jensen-Shannon divergence is the mean of these two KL divergences, giving $C(E, G)$:

$$\begin{aligned} C(E, G) &= 2\,\mathrm{D_{JS}}\left(P_{E\mathbf{X}} \,||\, P_{G\mathbf{Z}}\right) - \log 4 \\ &= \mathrm{D_{KL}}\left(P_{E\mathbf{X}} \,\middle|\middle|\, \tfrac{P_{E\mathbf{X}}+P_{G\mathbf{Z}}}{2}\right) + \mathrm{D_{KL}}\left(P_{G\mathbf{Z}} \,\middle|\middle|\, \tfrac{P_{E\mathbf{X}}+P_{G\mathbf{Z}}}{2}\right) - \log 4 \\ &= \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{X}}} \left[\mathbf{1}_{\left[E(\mathbf{x}) \in \hat{\Omega}_{\mathbf{Z}} \wedge G(E(\mathbf{x}))=\mathbf{x}\right]} \log f_{EG}(\mathbf{x}, E(\mathbf{x}))\right] + \\ &\quad \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}} \left[\mathbf{1}_{\left[G(\mathbf{z}) \in \hat{\Omega}_{\mathbf{X}} \wedge E(G(\mathbf{z}))=\mathbf{z}\right]} \log\left(1 - f_{EG}(G(\mathbf{z}), \mathbf{z})\right)\right] \square \end{aligned}$$

Appendix B Learning details

In this section we provide additional details on the BiGAN learning protocol summarized in Section 3.4. Goodfellow et al. (2014) found for GAN training that an objective in which the real and generated labels Y are swapped provides stronger gradient signal to G. We similarly observed in BiGAN training that an "inverse" objective Λ (with the same fixed point characteristics as V) provides stronger gradient signal to G and E, where

$$\Lambda(D, G, E) = \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{X}}}\big[\underbrace{\mathbb{E}_{\mathbf{z} \sim p_E(\cdot|\mathbf{x})}\left[\log\left(1 - D(\mathbf{x}, \mathbf{z})\right)\right]}_{\log(1-D(\mathbf{x},E(\mathbf{x})))}\big] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}}\big[\underbrace{\mathbb{E}_{\mathbf{x} \sim p_G(\cdot|\mathbf{z})}\left[\log D(\mathbf{x}, \mathbf{z})\right]}_{\log D(G(\mathbf{z}),\mathbf{z})}\big].$$

In practice, θ_G and θ_E are updated by moving in the positive gradient direction of this inverse objective $\nabla_{\theta_E,\theta_G}\Lambda$, rather than the negative gradient direction of the original objective.

We also observed that learning behaved similarly when all parameters θ_D, θ_G, θ_E were updated simultaneously at each iteration rather than alternating between θ_D updates and θ_G, θ_E updates, so we took the simultaneous updating (non-alternating) approach for computational efficiency. (For standard GAN training, simultaneous updates of θ_D, θ_G performed similarly well, so our standard GAN experiments also follow this protocol.)

APPENDIX C MODEL AND TRAINING DETAILS

In the following sections we present additional details on the models and training protocols used in the permutation-invariant MNIST and ImageNet evaluations presented in Section 4.

Optimization For unsupervised training of BiGANs and baseline methods, we use the Adam optimizer (Kingma & Ba, 2015) to compute parameter updates, following the hyperparameters (initial step size $\alpha = 2 \times 10^{-4}$, momentum $\beta_1 = 0.5$ and $\beta_2 = 0.999$) used by Radford et al. (2016). The step size α is decayed exponentially to $\alpha = 2 \times 10^{-6}$ starting halfway through training. The mini-batch size is 128. ℓ_2 weight decay of 2.5×10^{-5} is applied to all multiplicative weights in linear layers (but not to the learned bias β or scale γ parameters applied after batch normalization). Weights are initialized from a zero-mean normal distribution with a standard deviation of 0.02, with one notable exception: BiGAN discriminator weights that directly multiply $\mathbf{z}$ inputs to be added to spatial convolution outputs have initializations scaled by the convolution kernel size – e.g., for a 5×5 kernel, weights are initialized with a standard deviation of 0.5, 25 times the standard initialization.

Software & hardware We implement BiGANs and baseline feature learning methods using the *Theano* (Theano Development Team, 2016) framework, based on the convolutional GAN implementation provided by Radford et al. (2016). ImageNet transfer learning experiments (Section 4.3) use the *Caffe* (Jia et al., 2014) framework, per the Fast R-CNN (Girshick, 2015) and FCN (Long et al., 2015) reference implementations. Most computation is performed on an NVIDIA Titan X or Tesla K40 GPU.

C.1 PERMUTATION-INVARIANT MNIST

In all permutation-invariant MNIST experiments (Section 4.2), D, G, and E each consist of two hidden layers with 1024 units. The first hidden layer is followed by a non-linearity; the second is followed by (parameter-free) batch normalization (Ioffe & Szegedy, 2015) and a non-linearity. The second hidden layer in each case is the input to a linear prediction layer of the appropriate size. In D and E, a leaky ReLU (Maas et al., 2013) non-linearity with a “leak” of 0.2 is used; in G, a standard ReLU non-linearity is used. All models are trained for 400 epochs.

C.2 IMAGENET

In all ImageNet experiments (Section 4.3), the encoder E architecture follows AlexNet (Krizhevsky et al., 2012) through the fifth and last convolution layer (*conv5*), with local response normalization (LRN) layers removed and batch normalization (Ioffe & Szegedy, 2015) (including the learned scaling and bias) with leaky ReLU non-linearity applied to the output of each convolution at unsupervised training time. (For supervised evaluation, batch normalization is not used, and the pre-trained scale and bias is merged into the preceding convolution’s weights and bias.)

In most experiments, both the discriminator D and generator G architecture are those used by Radford et al. (2016), consisting of a series of four 5×5 convolutions (or “deconvolutions” – fractionally-strided convolutions – for the generator G) applied with 2 pixel stride, each followed by batch normalization and rectified non-linearity.

The sole exception is our discriminator baseline feature learning experiment, in which we let the discriminator D be the AlexNet variant described above. Generally, using AlexNet (or similar convnet architecture) as the discriminator D is detrimental to the visual fidelity of the resulting generated images, likely due to the relatively large convolutional filter kernel size applied to the input image, as well as the max-pooling layers, which explicitly discard information in the input. However, for fair comparison of the discriminator’s feature learning abilities with those of BiGANs, we use the same architecture as used in the BiGAN encoder.

Preprocessing To produce a data sample $\mathbf{x}$, we first sample an image from the database, and resize it proportionally such that its shorter edge has a length of 72 pixels. Then, a 64×64 crop is randomly selected from the resized image. The crop is flipped horizontally with probability $\frac{1}{2}$. Finally, the crop is scaled to $[-1, 1]$, giving the sample $\mathbf{x}$.

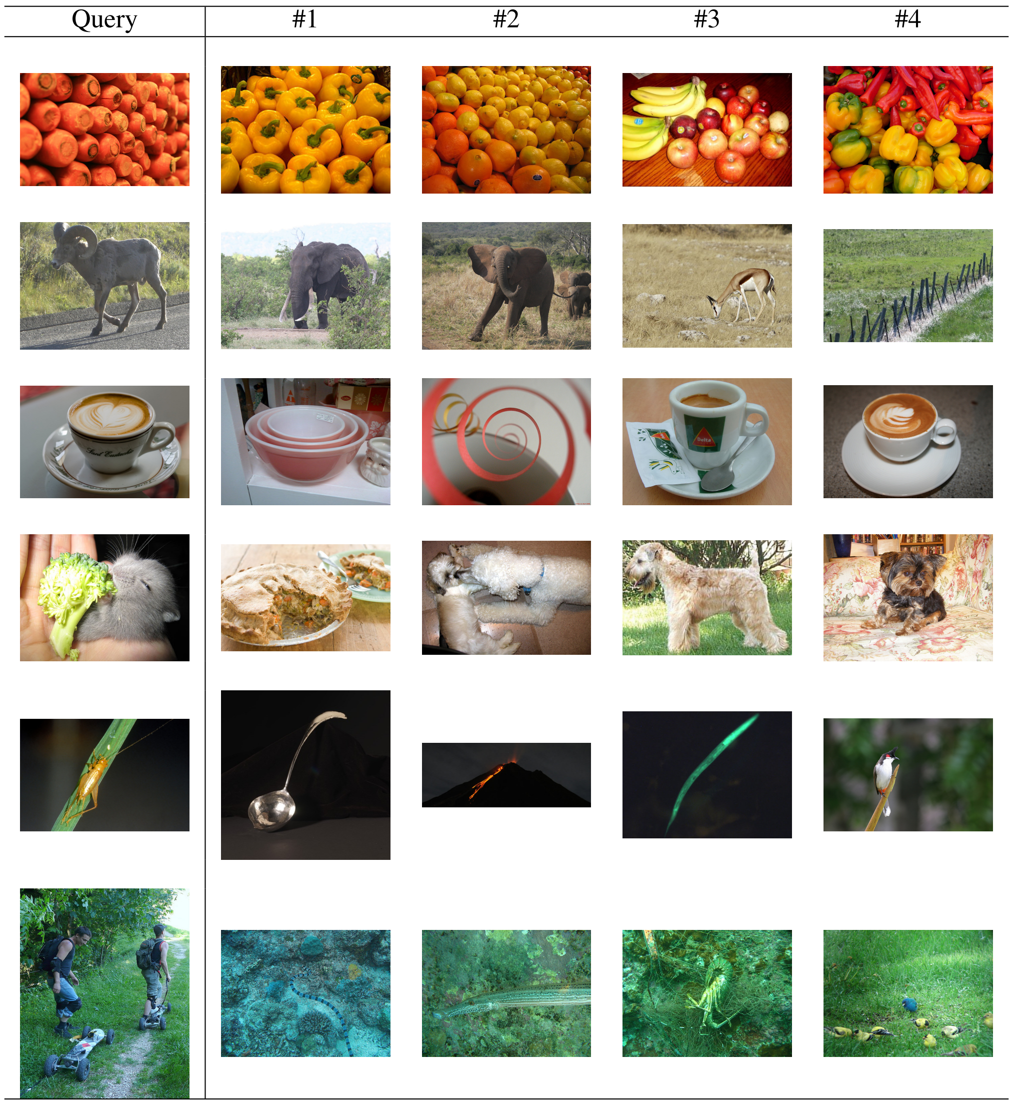

Figure 5: For the query images used in Krähenbühl et al. (2016) (left), nearest neighbors (by minimum cosine distance) from the ImageNet LSVRC (Russakovsky et al., 2015) training set in the *fc6* feature space of the ImageNet-trained BiGAN encoder E. (The *fc6* weights are set randomly; this space is a random projection of the learned *conv5* feature space.)

Timing A single epoch (one training pass over the 1.2 million images) of BiGAN training takes roughly 40 minutes on a Titan X GPU. Models are trained for 100 epochs, for a total training time of under 3 days.

Nearest neighbors In Figure 5 we present nearest neighbors in the feature space of the BiGAN encoder E learned in unsupervised ImageNet training.