Appendix

A. Main theorem and its proof

Theorem A.1 below, our main theorem, analyzes the extended KL-divergence for some $\beta \in (0.5, 1]$ defined as follows:

$$L_\beta(p) := \int (\beta p_*(x) + (1-\beta)p(x)) \ln \frac{\beta p_*(x) + (1-\beta)p(x)}{(1-\beta)p_*(x) + \beta p(x)} dx \; .$$

Theorem 2.1 above is a simplified version of Theorem A.1, and it can be obtained by setting $\beta = 1$ in Theorem A.1. The assumption of smooth and light-tailed p_* in Section 2.1 is precisely stated in Assumption A.1 below.

We first state the assumptions and then present Theorem A.1 and its proof.

A.1. Assumptions of Theorem A.1

A strong discriminator Given a set S_* of real data, a set S of generated data, and $\beta \in (0.5, 1]$ indicating that the probability $P(x^* \text{ is real}) = \beta$ for $x^* \in S_*$, and $P(x \text{ is real}) = 1 - \beta$ for $x \in S$, assume that we can obtain a strong discriminator D_β that solves the following weighed logistic regression problem:

$$D_\beta \approx \arg\min_{D'} \sum_{i: x_i \in S_* \cup S} w_i \left[\beta_i \ln(1 + \exp(-D'(x_i))) + (1-\beta_i) \ln(1 + \exp(D'(x_i)))\right]$$

$$(\beta_i, w_i) = \begin{cases} (\beta, \; 1/|S_*|) & \text{for } x_i \in S_* \quad \text{(real data)} \\ (1-\beta, \; 1/|S|) & \text{for } x_i \in S \quad \text{(generated data)} \end{cases}$$

Define a quantity $\mathcal{D}_\beta(x)$ by

$$\mathcal{D}_\beta(x) := \ln \frac{\beta p_*(x) + (1-\beta)p(x)}{(1-\beta)p_*(x) + \beta p(x)}$$

where p_* and p are the probability density functions of real data and generated data, respectively, and assume that there exists a positive number $B < \infty$ such that $|\mathcal{D}_\beta(x)| \le B$. Note that if we choose $\beta < 1$, then we can take $B = \ln(\beta/(1-\beta)) < \infty$, and the assumption on p_* and p can be relaxed from being both nonzero (with $\beta = 1$) to not being zero at the same time. However, in practice, one can simply take $\beta = 1$, and that is what was done in our experiments, because the practical behavior of choosing $\beta \approx 1$ is similar to $\beta = 1$.

When the number of given examples is sufficiently large, the standard statistical consistency theory of logistic regression implies that

$$D_\beta(x) \approx \mathcal{D}_\beta(x) \; .$$

Therefore, assume that the following ϵ-approximation condition is satisfied for a small $\epsilon > 0$:

$$\int p_*(x) \max(1, \|\nabla \ln p_*(x)\|) \left(|D_\beta(x) - \mathcal{D}_\beta(x)| + \left| e^{D_\beta(x)} - e^{\mathcal{D}_\beta(x)} \right| \right) dx \le \epsilon \; . \tag{6}$$

Smooth light-tailed p_* For convenience, we impose the following assumption.

Assumption A.1 *There are constants $c_0, h_0 > 0$ such that when $h \in (0, h_0)$, we have*

$$\int \sup_{\|g\| \le h} |p_*(x) + \nabla p_*(x)^\top g - p_*(x+g)| dx \le c_0 h^2,$$

$$\int \frac{\sup_{\|g\| \le h} |p_*(x+g) - p_*(x)|^2}{p_*(x)} dx \le c_0 h^2,$$

$$\int \|\nabla p_*(x)\| dx \le c_0.$$

The assumption says that p_* is a smooth density function with light tails. For example, common exponential distributions such as Gaussian distributions and mixtures of Gaussians all satisfy the assumption. It is worth mentioning that the

assumption is not truly needed for the algorithm. This is because an arbitrary distribution can always be approximated to an arbitrary precision by mixtures of Gaussians. Nevertheless, the assumption simplifies the statement of our analysis (Theorem A.1 below) because we do not have to deal with such approximations.

Also, to meet the assumption in practice, one can add a small Gaussian noise to every observed data point, as also noted by (Arjovsky & Bottou, 2017), but we empirically found that our method works without adding noise on image generation, which is good as we have one fewer meta-parameters.

A.2. Theorem A.1 and its proof

Theorem A.1 *Under the assumptions in Appendix A.1, let $g : \mathbb{R}^r \to \mathbb{R}^r$ be a continuously differentiable transformation such that $\|g(\cdot)\| \leq a$ and $\|\nabla g(\cdot)\| \leq b$. Let p be the probability density of a random variable X, and let p' be the probability density of the random variable X' such that $X' = X + \eta g(X)$ where $0 < \eta < \min(1/b, h_0/a)$. Then there exists a positive constant c such that for all $\epsilon > 0$:*

$$L_\beta(p') \leq L_\beta(p) - \eta \int p_*(x)u(x)\ g(x)^\top \nabla D_\beta(x)\ dx + c\eta^2 + c\eta\epsilon,$$

where $u(x) = \beta - (1-\beta)\exp(D_\beta(x))$.

Notation We use $\|\cdot\|$ to denote the vector 2-norm and the matrix spectral norm (the largest singular value of a matrix). Given a differentiable scalar function $h(x) : \mathbb{R}^r \to \mathbb{R}$, we use $\nabla h(x)$ to denote its gradient, which becomes a r-dimensional vector function. Given a differentiable function $g(x) : \mathbb{R}^r \to \mathbb{R}^r$, we use $\nabla g(x)$ to denote its Jacobi matrix and we use $\nabla \cdot g(x)$ to denote the divergence of $g(x)$, defined as

$$\nabla \cdot g(x) := \sum_{j=1}^{r} \frac{\partial g(x)}{\partial [x]_j},$$

where we use $[x]_j$ to denote the j-th component of x. We know that

$$\int \nabla \cdot w(x) dx = 0 \tag{7}$$

for all vector function $w(x)$ such that $w(\infty) = 0$.

Lemma A.1 *Assume that $g(x) : \mathbb{R}^r \to \mathbb{R}^r$ is a continuously differentiable transformation. Assume that $\|g(x)\| \leq a$ and $\|\nabla g(x)\| \leq b$, then as $\eta b < 1$, the inverse transformation $x = f^{-1}(x')$ of $x' = f(x) = x + \eta g(x)$ is unique.*

Moreover, consider transformation of random variables by $f^{-1}(\cdot)$. Define $\tilde{p}_*$ to be the associated probability density function after this transformation when the pdf before the transformation is p_*. Then for any $x \in \mathbb{R}^r$, we have:

$$\tilde{p}_*(x) = p_*(f(x))|\det(\nabla f(x))|. \tag{8}$$

Similarly, we have

$$p(x) = p'(f(x))|\det(\nabla f(x))|, \tag{9}$$

where p and p' are defined in Theorem A.1.

Proof Given x', define map $g'(x)$ as $g'(x) = x' - \eta g(x)$, then the assumption implies that $g'(x)$ is a contraction when $\eta b < 1$: $\|g'(x) - g'(x')\| \leq \eta b \|x - x'\|$. Therefore $g'(x)$ has a unique fixed point x, which leads to the inverse transformation $f^{-1}(x') = x$.

(8) and (9) follow from the standard density formula under transformation of variables. ■

Lemma A.2 *Under the assumptions of Lemma A.1, there exists a constant $c > 0$ such that*

$$|\det(\nabla f(x)) - (1 + \eta \nabla \cdot g(x))| \leq c\eta^2. \tag{10}$$

Proof

We note that

$$\nabla f(x) = I + \eta \nabla g(x).$$

Therefore

$$\det(\nabla f(x)) = 1 + \eta \nabla \cdot g(x) + \sum_{j \geq 2} \eta^j m_j(g(x)),$$

where $m_j(g)$ is a function of ∇g. Since ∇g is bounded, we obtain the desired formula. ■

Lemma A.3 *Under the assumptions of Lemma A.1, and assume that Assumption A.1 holds, then there exists a constant $c > 0$ such that*

$$\int \left|\tilde{p}_*(x) - (p_*(x) + \eta p_*(x) \nabla \cdot g(x) + \eta \nabla p_*(x)^\top g(x))\right| dx \leq c\eta^2. \tag{11}$$

and

$$\int \frac{(\tilde{p}_*(x) - p_*(x))^2}{p_*(x)} dx \leq c\eta^2. \tag{12}$$

Proof Using the algebraic inequality

$$\begin{aligned}
&\left|p_*(f(x))|\det(\nabla f(x))| - (p_*(x) + \eta p_*(x) \nabla \cdot g(x) + \eta \nabla p_*(x)^\top g(x))\right| \\
\leq &\left|p_*(f(x)) - (p_*(x) + \eta \nabla p_*(x)^\top g(x))\right| \, |\det(\nabla f(x))| \\
&+ \left|(p_*(x) + \eta \nabla p_*(x)^\top g(x))\right| \, \left|(1 + \eta \nabla \cdot g(x)) - |\det(\nabla f(x))|\right| \\
&+ \eta^2 \left|\nabla \cdot g(x) \; \nabla p_*(x)^\top g(x))\right|,
\end{aligned}$$

and using $\tilde{p}_*(x) = p_*(f(x))|\det(\nabla f(x))|$ from (8), we obtain

$$\begin{aligned}
&\int \left|\tilde{p}_*(x) - (p_*(x) + \eta p_*(x) \nabla \cdot g(x) + \eta \nabla p_*(x)^\top g(x))\right| dx \\
\leq &\underbrace{\int \left|p_*(f(x)) - (p_*(x) + \eta \nabla p_*(x)^\top g(x))\right| \, |\det(\nabla f(x))| dx}_{A_0} \\
&+ \underbrace{\int \left|(p_*(x) + \eta \nabla p_*(x)^\top g(x))\right| \, \left|(1 + \eta \nabla \cdot g(x)) - |\det(\nabla f(x))|\right| \, dx}_{B_0} \\
&+ \eta^2 \underbrace{\int \left|\nabla \cdot g(x) \; \nabla p_*(x)^\top g(x))\right| dx}_{C_0} \\
\leq &c\eta^2
\end{aligned}$$

for some constant $c > 0$, which proves (11). The last inequality uses the following facts.

$$A_0 = \int \left|p_*(f(x)) - (p_*(x) + \eta \nabla p_*(x)^\top g(x))\right| \, O(1) dx = O(\eta^2),$$

where the first equality follows from the boundedness of g and ∇g, and the second equality follows from the first inequality of Assumption A.1.

$$B_0 = \int \left|(p_*(x) + \eta \nabla p_*(x)^\top g(x))\right| \, O(\eta^2) \, dx = O(\eta^2),$$

where the first equality follows from (10), and the second equality follows from the third equality of Assumption A.1.

$$C_0 = \int \|\nabla p_*(x)\| O(1) dx = O(1),$$

where the first equality follows from the boundedness of g and ∇g, and the second equality follows from the third equality of Assumption A.1.

Moreover, using (8), we obtain

$$|\tilde{p}_*(x) - p_*(x)| \leq |p_*(f(x)) - p_*(x)| \, |\det(\nabla f(x))| + p_*(x)||\det(\nabla f(x))| - 1|.$$

Therefore

$$\begin{aligned}
&\int \frac{(\tilde{p}_*(x) - p_*(x))^2}{p_*(x)} dx \\
\leq & 2 \int \frac{(p_*(f(x)) - p_*(x))^2 |\det(\nabla f(x))|^2 + p_*(x)^2 (|\det(\nabla f(x))| - 1)^2}{p_*(x)} dx \leq c\eta^2
\end{aligned}$$

for some $c > 0$, which proves (12). The second inequality follows from the second inequality of Assumption A.1, and the boundedness of $|\det(\nabla f(x))|$, and the fact that $||\det(\nabla f(x))| - 1| = O(\eta)$ from (10). ■

Proof of Theorem A.1

In the following integration, with a change of variable from x to x' using $x' = f(x)$ as in Lemma A.1, we obtain

$$\begin{aligned}
&\int (\beta p_*(x') + (1-\beta)p'(x')) \ln \frac{\beta p_*(x') + (1-\beta)p'(x')}{(1-\beta)p_*(x') + \beta p'(x')} dx' \\
= &\int (\beta p_*(f(x)) + (1-\beta)p'(f(x))) \ln \frac{\beta p_*(f(x)) + (1-\beta)p'(f(x))}{(1-\beta)p_*(f(x)) + \beta p'(f(x))} |\det(\nabla f(x))| dx \\
= &\int (\beta \tilde{p}_*(x) + (1-\beta)p(x)) \ln \frac{\beta \tilde{p}_*(x) + (1-\beta)p(x)}{(1-\beta)\tilde{p}_*(x) + \beta p(x)} dx,
\end{aligned}$$

where the first inequality is basic calculus, and the second inequality uses (8) and (9).

It follows that

$$\begin{aligned}
L_\beta(p') = &\int (\beta p_*(x') + (1-\beta)p'(x')) \ln \frac{\beta p_*(x') + (1-\beta)p'(x')}{(1-\beta)p_*(x') + \beta p'(x')} dx' \\
= &\int (\beta \tilde{p}_*(x) + (1-\beta)p(x)) \ln \frac{\beta \tilde{p}_*(x) + (1-\beta)p(x)}{(1-\beta)\tilde{p}_*(x) + \beta p(x)} dx \\
= &A_1 + B_1 + C_1,
\end{aligned}$$

where A_1, B_1, and C_1 are defined as follows.

$$\begin{aligned}
A_1 = &\int (\beta \tilde{p}_*(x) + (1-\beta)p(x)) \ln \frac{\beta p_*(x) + (1-\beta)p(x)}{(1-\beta)p_*(x) + \beta p(x)} dx \\
= &\int (\beta p_*(x) + (1-\beta)p(x)) \ln \frac{\beta p_*(x) + (1-\beta)p(x)}{(1-\beta)p_*(x) + \beta p(x)} dx \\
&+ \eta\beta \int (p_*(x) \nabla \cdot g(x) + \nabla p_*(x)^\top g(x)) \ln \frac{\beta p_*(x) + (1-\beta)p(x)}{(1-\beta)p_*(x) + \beta p(x)} dx + O(\eta^2) \\
= &L_\beta(p) + \beta\eta \int \nabla \cdot (p_*(x) g(x)) \, \mathcal{D}_\beta(x) dx + O(\eta^2) \\
= &L_\beta(p) + \beta\eta \int \nabla \cdot (p_*(x) g(x)) \, D_\beta(x) dx + O(\eta\epsilon + \eta^2) \\
= &L_\beta(p) - \beta\eta \int p_*(x) g(x)^\top \nabla D_\beta(x) dx + O(\eta\epsilon + \eta^2),
\end{aligned}$$

where the second equality uses (11) and the fact that $B < \infty$ in the statement of the theorem. The fourth equality uses the ϵ-approximation condition (6) of the assumption. The last equality uses integration by parts and (7).

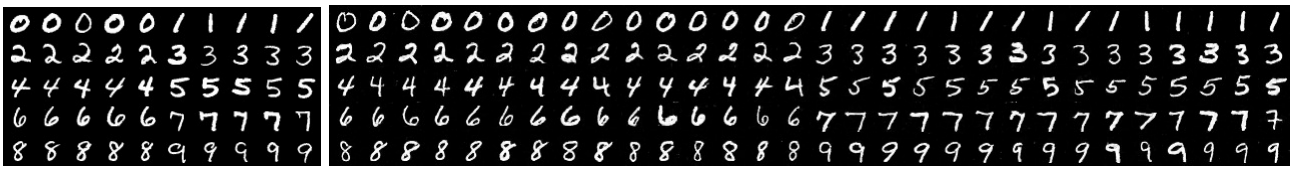
$$
\begin{aligned}
B_1 =& \int (\beta\tilde{p}_*(x) + (1-\beta)p(x)) \ln \frac{\beta\tilde{p}_*(x) + (1-\beta)p(x)}{\beta p_*(x) + (1-\beta)p(x)} dx \\
=& \int (\beta\tilde{p}_*(x) + (1-\beta)p(x)) \ln \left(1 + \beta \frac{\tilde{p}_*(x) - p_*(x)}{\beta p_*(x) + (1-\beta)p(x)}\right) dx \\
\leq& \beta \int (\beta\tilde{p}_*(x) + (1-\beta)p(x)) \frac{\tilde{p}_*(x) - p_*(x)}{\beta p_*(x) + (1-\beta)p(x)} dx \\
=& \beta^2 \int \frac{(\tilde{p}_*(x) - p_*(x))^2}{\beta p_*(x) + (1-\beta)p(x)} dx = O(\eta^2),
\end{aligned}
$$

where the inequality uses $\ln(1+\delta) \leq \delta$. The last equality uses (12).

$$
\begin{aligned}
C_1 =& \int (\beta\tilde{p}_*(x) + (1-\beta)p(x)) \ln \frac{(1-\beta)p_*(x) + \beta p(x)}{(1-\beta)\tilde{p}_*(x) + \beta p(x)} dx \\
=& \int (\beta\tilde{p}_*(x) + (1-\beta)p(x)) \ln \left(1 + (1-\beta) \frac{p_*(x) - \tilde{p}_*(x)}{(1-\beta)\tilde{p}_*(x) + \beta p(x)}\right) dx \\
\leq& (1-\beta) \int (\beta\tilde{p}_*(x) + (1-\beta)p(x)) \frac{p_*(x) - \tilde{p}_*(x)}{(1-\beta)\tilde{p}_*(x) + \beta p(x)} dx \\
=& (1-\beta) \int (\beta\tilde{p}_*(x) + (1-\beta)p(x)) \frac{p_*(x) - \tilde{p}_*(x)}{(1-\beta)p_*(x) + \beta p(x)} dx \\
&+ (1-\beta)^2 \int (\beta\tilde{p}_*(x) + (1-\beta)p(x)) \frac{(p_*(x) - \tilde{p}_*(x))^2}{((1-\beta)\tilde{p}_*(x) + \beta p(x))((1-\beta)p_*(x) + \beta p(x))} dx \\
\underset{(a)}{=}& (1-\beta) \int (\beta\tilde{p}_*(x) + (1-\beta)p(x)) \frac{p_*(x) - \tilde{p}_*(x)}{(1-\beta)p_*(x) + \beta p(x)} dx + O(\eta^2) \\
\underset{(b)}{=}& -(1-\beta) \int (\beta p_*(x) + (1-\beta)p(x)) \frac{\eta p_*(x) \nabla \cdot g(x) + \eta \nabla p_*(x)^\top g(x)}{(1-\beta)p_*(x) + \beta p(x)} dx + O(\eta^2) \\
=& -(1-\beta)\eta \int (p_*(x) \nabla \cdot g(x) + \nabla p_*(x)^\top g(x)) \exp(\mathcal{D}_\beta(x)) dx + O(\eta^2) \\
\underset{(c)}{=}& -(1-\beta)\eta \int (p_*(x) \nabla \cdot g(x) + \nabla p_*(x)^\top g(x)) \exp(D_\beta(x)) dx + O(\eta\epsilon + \eta^2) \\
=& -(1-\beta)\eta \int (\nabla \cdot (p_*(x) g(x))) \exp(D_\beta(x)) dx + O(\eta\epsilon + \eta^2) \\
=& \eta(1-\beta) \int p_*(x) g(x)^\top \nabla \exp(D_\beta(x)) dx + O(\eta\epsilon + \eta^2),
\end{aligned}
$$

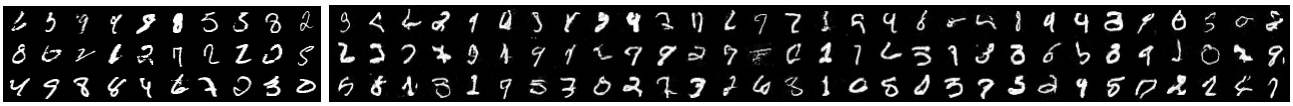
where the first inequality uses $\ln(1+\delta) \leq \delta$. The equality (a) uses (12). The equality (b) uses (11). The equality (c) uses the ϵ-approximation condition (6). The last equality uses integration by parts and (7).

By combining the estimates of A_1, B_1, and C_1, we obtain the desired bound.



(a) Real images (b) Generated by xICFG (convolutional)

Figure 12. 'Best' digits. MNIST. For each digit, showing images with the highest probabilities among 1000 images that were either (a) randomly chosen from real data or (b) generated by xICFG.



(a) Real images (b) Generated by xICFG (convolutional)

Figure 13. 'Worst' digits. MNIST. Images with the highest entropy among 1000 images that were either (a) randomly chosen from real data or (b) generated by xICFG. Some of the generated images in (b) are hard to tell what digits they are, but so are some of the real images in (a).

(a) Real images (b) Generated by xICFG (convolutional)

Figure 14. 'Best' digits. SVHN. For each digit, showing images with the highest probabilities among 1000 images that were eigher (a) randomly chosen from real data or (b) generated by xICFG.

(a) Real images (b) Generated by xICFG (convolutional)

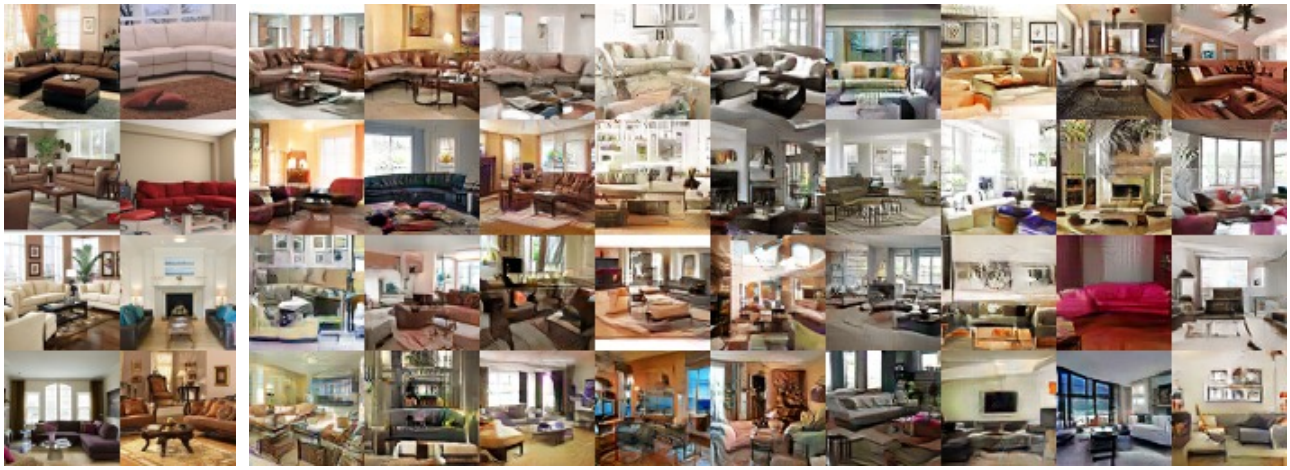
Figure 15. 'Worst' digits. SVHN. Images with the highest entropy among 1000 images that were either (a) randomly chosen from real data or (b) generated by xICFG. Some of the generated images in (b) are hard to tell what digits they are, but so are some of the real images in (a).

B. Image examples

We have shown random samples of generated images, and here we take a more focused approach. We generate 1000 images by xICFG trained in the settings of Figure 2 and show (roughly) the best and worst images among them. Similar to the inception score, the 'goodness' of images are measured by the confidence of a classifier, e.g., a image that a classifier assigns a high probability of being a "bedroom" is considered to be a good bedroom image. The worst images are those with the highest entropy values. In Figures 12–21, we compare real images and generated images side by side that were chosen by the same procedure from a random sample of 1000 real images or 1000 generated images (generated from one sequence of random inputs), respectively.

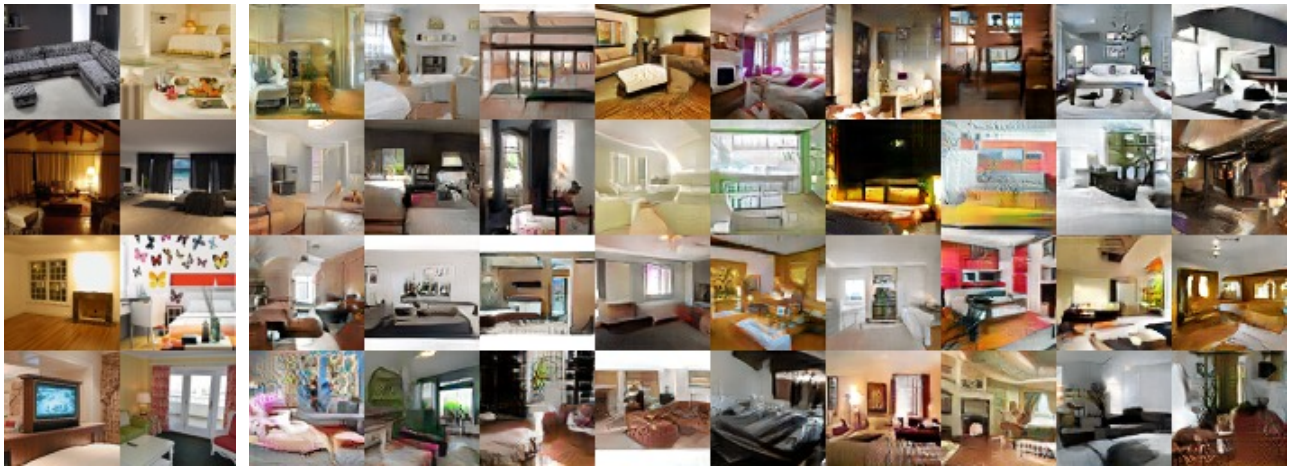
(a) Real images. (b) Generated by xICFG (4-block ResNet)

Figure 16. Bedrooms 'best' among 1000 (LSUN BR+LR). Predicted by a classifier to be "bedroom" with the highest probabilities among 1000 images that were either (a) randomly chosen from real data or (b) generated by xICFG.



(a) Real images. (b) Generated by xICFG (4-block ResNet)

Figure 17. Living rooms 'best' among 1000 (LSUN BR+LR). Predicted by a classifier to be "living room" with the highest probabilities among 1000 images that were either (a) randomly chosen from real data or (b) generated by xICFG.



(a) Real images. (b) Generated by xICFG (4-block ResNet)

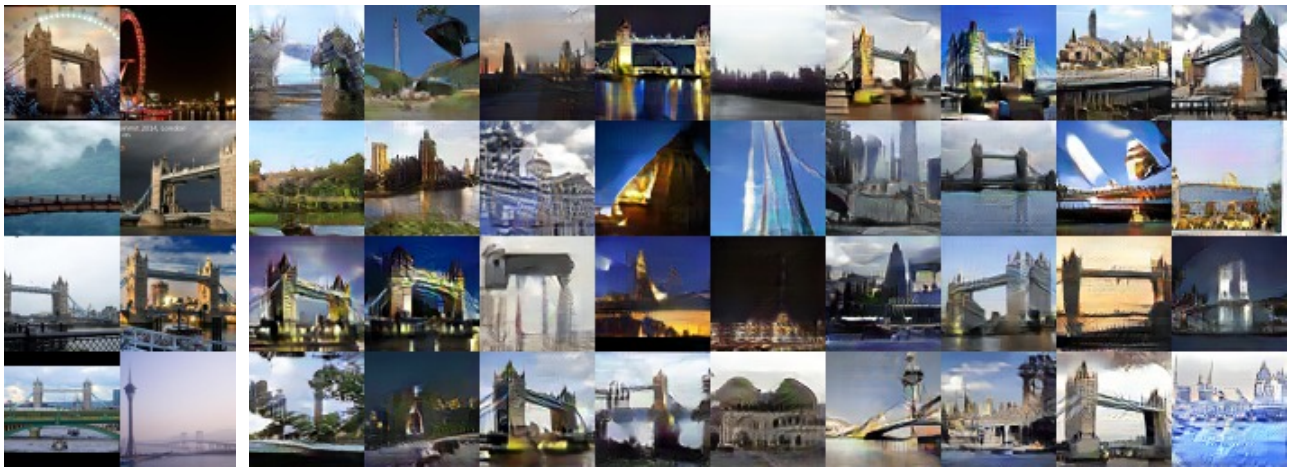
Figure 18. Bedrooms/living rooms 'worst' among 1000 (LSUN BR+LR). Images with the highest entropy among 1000 images that were either (a) randomly chosen from real data or (b) generated by xICFG. The generated images in (b) could be either of relatively low quality or depicting hard-to-tell rooms as the real images in (a) do.

(a) Real images. (b) Generated by xICFG (4-block ResNet)

Figure 19. Towers 'best' among 1000 (LSUN T+B). Predicted by a classifier to be "tower" with the highest probabilities among 1000 images that were either (a) randomly chosen from real data or (b) generated by xICFG.

(a) Real images. (b) Generated by xICFG (4-block ResNet)

Figure 20. Bridges 'best' among 1000 (LSUN T+B). Predicted by a classifier to be "bridge" with the highest probabilities among 1000 images that were either (a) randomly chosen from real data or (b) generated by xICFG.



(a) Real images. (b) Generated by xICFG (4-block ResNet)

Figure 21. Towers/bridges 'worst' among 1000 (LSUN T+B). Images with the highest entropy among 1000 images that were either (a) randomly chosen from real data or (b) generated by xICFG. The generated images in (b) could be either of relatively low quality or depicting hard-to-tell objects as the real images in (a) do.

C. Details of experimental setup

C.1. Network architectures

The network definitions below include batch normalization layers. Note that to experiment with WGANgp, we explored the options with the batch normalization layers being removed or partially removed, as described in Section 4.1.

C.1.1. MNIST AND SVHN IN FIGURE 2

The convolutional architectures used for MNIST and SVHN are an extension of DCGAN, inserting 1×1 convolution layers.

Approximator/Generator

1	Projection
2	ReLU
3	Transposed conv, 5×5, stride 2
4	BatchNorm
5	ReLU
6	Convolution, 1×1, stride 1
7	BatchNorm
8	ReLU
9	Transposed conv, 5×5, stride 2
10	tanh

Repeat 2–7 twice.

Discriminator

1	Convolution, 5×5, stride 2
2	LeakyReLU
3	Convolution, 5×5, stride 2
4	BatchNorm
5	LeakyReLU
6	Convolution, 1×1, stride 1
7	BatchNorm
8	LeakyReLU
9	Flatten
10	Linear

Repeat 3–8 twice.

For the discriminator, start with 32 (MNIST) or 64 (SVHN) feature maps and double it at downsampling except for the first downsampling. For the approximator/generator, start with 128 (MNIST) or 256 (SVHN) and halve it at upsampling except for the last upsampling.

C.1.2. LSUN IN FIGURE 2

The convolutional architecture used for LSUN is a simplification of a residual network found at https://github.com/igul222/improved_wgan_training, reducing the number of batch normalization layers for speedup, removing some irregularity, and so forth. Both the approximator/generator and discriminator are a residual network with four convolution blocks.

Approximator/Generator

1	Projection
2	ReLU
3	Upsampling ($\times2$), nearest
4	Convolution, 3×3, stride 1
5	ReLU
6	Convolution, 3×3, stride 1
7	BatchNorm
8	ReLU
9	Convolution, 3×3, stride 1
10	tanh

Repeat 2–7 four times.

Discriminator

1	ReLU (omitted in the 1st block)
2	Convolution, 3×3, stride 1
3	ReLU
4	Convolution, 3×3, stride 1
5	BatchNorm
6	Downsampling (/2), mean
7	ReLU
8	Flatten
9	Linear

Repeat 1–6 four times.

2–7 of the approximator/generator and 1–6 of the discriminator are convolution blocks with a shortcut connecting the beginning to the end. In the approximator/generator, the numbers of feature maps are 512 (produced by the projection layer) and 256, 256, 128, 128, 64, 64, 64, and 64 (produced by the convolution layers). In the discriminator, the numbers of feature maps produced by the convolution layers are 64, 64, 64, 128, 128, 256, 256, and 512.

C.2. Details of experimental settings for xICFG

The network weights were initialized by the Gaussian with mean 0 and standard deviation 0.01.

The rmsprop learning rate for xICFG (for updating the discriminator and the approximator) was fixed to 0.0001 across all the

datasets when the approximator was fully-connected. Although 0.0001 worked well also for the convolutional approximator cases, these cases turned out to tolerate and benefit from a more aggressive learning rate resulting in faster training; hence, it was fixed to 0.00025 across all the datasets. Additionally, if we keep training long enough, the discriminator may eventually overfit as also noted on WGANgp in (Gulrajani et al., 2017). It may be useful to reduce the learning rate (e.g., by multiplying 0.1) towards the end of training if the onset of discriminator overfit needs to be delayed.

As shown in Table 1, the discriminator update frequency U was fixed to 1. However, it is worth mentioning that stable training has been observed with a relatively wide range of U including U=25. The choice of reporting the results with U=1 was due to its pragmatic advantage – training tends to be faster with a smaller U as it leads to more frequent updates of a generator.

To choose η used for generator update $G_{t+1}(z) = G_t(z) + \eta \nabla D(G_t(z))$, we tried some of $\{0.1, 0.25, 0.5, 1, 2.5\}$ (not all as we tried only promising ones) for each configuration, following the meta-parameter selection protocol described in Section 4.1. Typically, multiple values were found to work well, and the table below shows the chosen values.

	MNIST	SVHN	BR+LR	T+B
convolutional (Fig.2)	1	0.25	1	1
conv. no batch norm (Fig.2)	–	0.5	2.5	–
fully-connected (Fig.3)	0.1	0.25	0.5	0.5

In general, similar to the SGD learning rate, a larger η leads to faster training, but a too large value would break training. A too small value should be avoided since stable training requires a generator to make sufficient progress before each approximator update.