Adaptive Density Estimation for Generative Models

Thomas Lucas, Konstantin Shmelkov, Cordelia Schmid, Karteek Alahari, Jakob Verbeek

▶ **To cite this version:**

Thomas Lucas, Konstantin Shmelkov, Cordelia Schmid, Karteek Alahari, Jakob Verbeek. Adaptive Density Estimation for Generative Models. NeurIPS 2019 - Thirty-third Conference on Neural Information Processing Systems, Dec 2019, Vancouver, Canada. pp.11993-12003. hal-01886285v5

HAL Id: hal-01886285
https://hal.science/hal-01886285v5

Submitted on 14 Feb 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Adaptive Density Estimation for Generative Models

Thomas Lucas[*]
Inria[†]

Konstantin Shmelkov[*,‡]
Noah's Ark Lab, Huawei

Cordelia Schmid
Inria[†]

Karteek Alahari
Inria[†]

Jakob Verbeek
Inria[†]

Abstract

Unsupervised learning of generative models has seen tremendous progress over recent years, in particular due to generative adversarial networks (GANs), variational autoencoders, and flow-based models. GANs have dramatically improved sample quality, but suffer from two drawbacks: (i) they mode-drop, *i.e.*, do not cover the full support of the train data, and (ii) they do not allow for likelihood evaluations on held-out data. In contrast, likelihood-based training encourages models to cover the full support of the train data, but yields poorer samples. These mutual shortcomings can in principle be addressed by training generative latent variable models in a hybrid adversarial-likelihood manner. However, we show that commonly made parametric assumptions create a conflict between them, making successful hybrid models non trivial. As a solution, we propose to use deep invertible transformations in the latent variable decoder. This approach allows for likelihood computations in image space, is more efficient than fully invertible models, and can take full advantage of adversarial training. We show that our model significantly improves over existing hybrid models: offering GAN-like samples, IS and FID scores that are competitive with fully adversarial models, and improved likelihood scores.

1 Introduction

Successful recent generative models of natural images can be divided into two broad families, which are trained in fundamentally different ways. The first is trained using likelihood-based criteria, which ensure that all training data points are well covered by the model. This category includes variational autoencoders (VAEs) [25, 26, 39, 40], autoregressive models such as PixelCNNs [46, 53], and flow-based models such as Real-NVP [9, 20, 24]. The second category is trained based on a signal that measures to what extent (statistics of) samples from the model can be distinguished from (statistics of) the training data, *i.e.*, based on the quality of samples drawn from the model. This is the case for generative adversarial networks (GANs) [2, 15, 22], and moment matching methods [28].

Despite tremendous recent progress, existing methods exhibit a number of drawbacks. Adversarially trained models such as GANs do not provide a density function, which poses a fundamental problem as it prevents assessment of how well the model fits held out and training data. Moreover, adversarial models typically do not allow to infer the latent variables that underlie observed images. Finally, adversarial models suffer from mode collapse [2], *i.e.*, they do not cover the full support of the training data. Likelihood-based model on the other hand are trained to put probability mass on all elements of the training set, but over-generalise and produce samples of substantially inferior quality as compared to adversarial models. The models with the best likelihood scores on held-out data are autoregressive models [35], which suffer from the additional problem that they are extremely

[*] The authors contributed equally. [†] Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France. [‡] Work done while at Inria.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

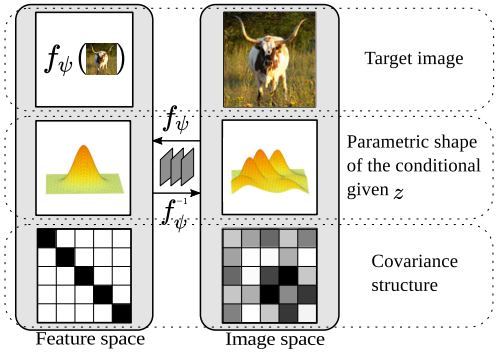


Figure 1: An invertible non-linear mapping f_ψ maps an image x to a vector $f_\psi(x)$ in feature space. f_ψ is trained to adapt to modelling assumptions made by a trained density p_θ in feature space. This induces full covariance structure and a non-Gaussian density.

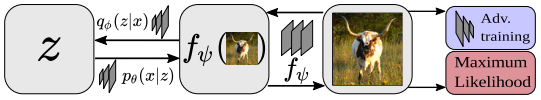


Figure 2: Variational inference is used to train a latent variable generative model in feature space. The invertible mapping f_ψ maps back to image space, where adversarial training can be performed together with MLE.

Figure 3: Our model yields compelling samples while the optimization of likelihood ensures coverage of all modes in the training support and thus sample diversity, here on LSUN churches (64×64).

inefficient to sample from [38], since images are generated pixel-by-pixel. The sampling inefficiency makes adversarial training of such models prohibitively expensive.

In order to overcome these shortcomings, we seek to design a model that (i) generates high-quality samples typical of adversarial models, (ii) provides a likelihood measure on the entire image space, and (iii) has a latent variable structure to enable efficient sampling and to permit adversarial training. Additionally we show that, (iv) a successful hybrid adversarial-likelihood paradigm requires going beyond simplifying assumptions commonly made with likelihood based latent variable models. These simplifying assumptions on the conditional distribution on data x given latents z, $p(x|z)$, include full independence across the dimensions of x and/or simple parametric forms such as Gaussian [25], or use fully invertible networks [9, 24]. These assumptions create a conflict between achieving high sample quality and high likelihood scores on held-out data. Autoregressive models, such as pixelCNNs [46, 53], do not make factorization assumptions, but are extremely inefficient to sample from. As a solution, we propose learning a non-linear invertible function f_ψ between the image space and an abstract feature space, as illustrated in Figure 1. Training a model with full support in this feature space induces a model in the image space that does not make Gaussianity, nor independence assumptions in the conditional density $p(x|z)$. Trained by MLE, f_ψ adapts to modelling assumptions made by p_θ so we refer to this approach as "adaptive density estimation".

We experimentally validate our approach on the CIFAR-10 dataset with an ablation study. Our model significantly improves over existing hybrid models, producing GAN-like samples, and IS and FID scores that are competitive with fully adversarial models, see Figure 3. At the same time, we obtain likelihoods on held-out data comparable to state-of-the-art likelihood-based methods which requires covering the full support of the dataset. We further confirm these observations with quantitative and qualitative experimental results on the STL-10, ImageNet and LSUN datasets.

2 Related work

Mode-collapse in GANs has received considerable attention, and stabilizing the training process as well as improved and bigger architectures have been shown to alleviate this issue [2, 17, 37]. Another line of work focuses on allowing the discriminator to access batch statistics of generated images, as pioneered by [22, 45], and further generalized by [29, 32]. This enables comparison of distributional statistics by the discriminator rather than only individual samples. Other approaches to encourage diversity among GAN samples include the use of maximum mean discrepancy [1], optimal transport [47], determinental point processes [14] and Bayesian formulations of adversarial training [42] that allow model parameters to be sampled. In contrast to our work, these models lack an inference network, and do not define an explicit density over the full data support.

An other line of research has explored inference mechanisms for GANs. The discriminator of BiGAN [10] and ALI [12], given pairs (x, z) of images and latents, predict if z was encoded from a real image, or if x was decoded from a sampled z. In [52] the encoder and the discriminator are collapsed into one network that encodes both real images and generated samples, and tries to

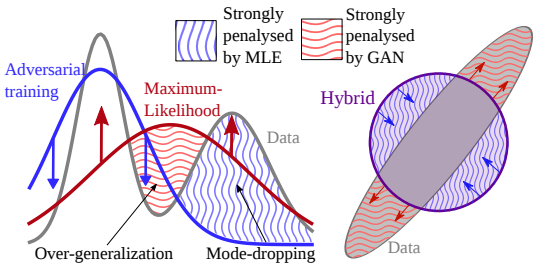


Figure 4: (Left) Maximum likelihood training pulls probability mass towards high-density regions of the data distribution, while adversarial training pushes mass out of low-density regions. (Right) Independence assumptions become a source of conflict in a joint training setting, making hybrid training non-trivial.

spread their posteriors apart. In [6] a symmetrized KL divergence is approximated in an adversarial setup, and uses reconstruction losses to improve the correspondence between reconstructed and target variables for x and z. Similarly, in [41] a discriminator is used to replace the KL divergence term in the variational lower bound used to train VAEs with the density ratio trick. In [33] the KL divergence term in a VAE is replaced with a discriminator that compares latent variables from the prior and the posterior in a more flexible manner. This regularization is more flexible than the standard KL divergence. The VAE-GAN model [27] and the model in [21] use the intermediate feature maps of a GAN discriminator and of a classifier respectively, as target space for a VAE. Unlike ours, these methods do not define a likelihood over the image space.

Likelihood-based models typically make modelling assumptions that conflict with adversarial training, these include strong factorization and/or Gaussianity. In our work we avoid these limitations by learning the shape of the conditional density on observed data given latents, $p(x|z)$, beyond fully factorized Gaussian models. As in our work, Flow-GAN [16] also builds on invertible transformations to construct a model that can be trained in a hybrid adversarial-MLE manner, see Figure 2.However, Flow-GAN does not use efficient non-invertible layers we introduce, and instead relies entirely on invertible layers. Other approaches combine autoregressive decoders with latent variable models to go beyond typical parametric assumptions in pixel space [7, 18, 31]. They, however, are not amenable to adversarial training due to the prohibitively slow sequential pixel sampling.

3 Preliminaries on MLE and adversarial training

Maximum-likelihood and over-generalization. The de-facto standard approach to train generative models is maximum-likelihood estimation. It maximizes the probability of data sampled from an unknown data distribution p^* under the model p_θ w.r.t. the model parameters θ. This is equivalent to minimizing the Kullback-Leibler (KL) divergence, $\mathcal{D}_{\mathrm{KL}}(p^*||p_\theta)$, between p^* and p_θ. This yields models that tend to cover all the modes of the data, but put mass in spurious regions of the target space; a phenomenon known as "over-generalization" or "zero-avoiding" [4], and manifested by unrealistic samples in the context of generative image models, see Figure 4.Over-generalization is inherent to the optimization of the KL divergence oriented in this manner. Real images are sampled from p^*, and p_θ is explicitly optimized to cover all of them. The training procedure, however, does not sample from p_θ to evaluate the quality of such samples, ideally using the inaccessible $p^*(x)$ as a score. Therefore p_θ may put mass in spurious regions of the space without being heavily penalized. We refer to this kind of training procedure as "coverage-driven training" (CDT). This optimizes a loss of the form $\mathcal{L}_{\mathrm{C}}(p_\theta) = \int_x p^*(x) s_c(x, p_\theta)\ \mathrm{d}x$, where $s_c(x, p_\theta) = \ln p_\theta(x)$ evaluates how well a sample x is covered by the model. Any score s_c that verifies: $\mathcal{L}_{\mathrm{C}}(p_\theta) = 0 \iff p_\theta = p^*$, is equivalent to the log-score, which forms a justification for MLE on which we focus.

Explicitly evaluating sample quality is redundant in the regime of unlimited model capacity and training data. Indeed, putting mass on spurious regions takes it away from the support of p^*, and thus reduces the likelihood of the training data. In practice, however, datasets and model capacity are finite, and models *must* put mass outside the finite training set in order to generalize. The maximum likelihood criterion, by construction, only measures *how much* mass goes off the training data, not *where* it goes. In classic MLE, generalization is controlled in two ways: (i) inductive bias, in the form of model architecture, controls *where* the off-dataset mass goes, and (ii) regularization controls to which extent this happens. An adversarial loss, by considering samples of the model p_θ, can provide a second handle to evaluate and control where the off-dataset mass goes. In this sense, and in contrast to model architecture design, an adversarial loss provides a "trainable" form of inductive bias.

Adversarial models and mode collapse. Adversarially trained models produce samples of excellent quality. As mentioned, their main drawbacks are their tendency to "mode-drop", and the lack of measure to assess mode-dropping, or their performance in general. The reasons for this are two-fold. First, defining a valid likelihood requires adding volume to the low-dimensional manifold learned by GANs to define a density under which training and test data have non-zero density. Second, computing the density of a data point under the defined probability distribution requires marginalizing out the latent variables, which is not trivial in the absence of an efficient inference mechanism. When a human expert subjectively evaluates the quality of generated images, samples from the model are compared to the expert's implicit approximation of p^*. This type of objective may be formalized as $\mathcal{L}_{\text{Q}}(p_\theta) = \int_x p_\theta(x) s_q(x, p^*)\ \mathrm{d}x$, and we refer to it as "quality-driven training" (QDT). To see that GANs [15] use this type of training, recall that the discriminator is trained with the loss $\mathcal{L}_{\text{GAN}} = \int_x p^*(x) \ln D(x) + p_\theta(x) \ln(1 - D(x))\ \mathrm{d}x$. It is easy to show that the optimal discriminator equals $D^*(x) = p^*(x)/(p^*(x) + p_\theta(x))$. Substituting the optimal discriminator, $\mathcal{L}_{\text{GAN}}$ equals the Jensen-Shannon divergence,

$$\mathcal{D}_{\text{JS}}(p^*||p_\theta) = \frac{1}{2}\mathcal{D}_{\text{KL}}(p^*||\frac{1}{2}(p_\theta + p^*)) + \frac{1}{2}\mathcal{D}_{\text{KL}}(p_\theta||\frac{1}{2}(p_\theta + p^*)), \tag{1}$$

up to additive and multiplicative constants [15]. This loss, approximated by the discriminator, is symmetric and contains two KL divergence terms. Note that $\mathcal{D}_{\text{KL}}(p^*||\frac{1}{2}(p_\theta + p^*))$ is an integral on p^*, so *coverage driven*. The term that approximates it in $\mathcal{L}_{\text{GAN}}$, *i.e.*, $\int_x p^*(x) \ln D(x)$, is however independent from the generative model, and disappears when differentiating. Therefore, it cannot be used to perform coverage-driven training, and the generator is trained to minimize $\ln(1 - D(G(z)))$ (or to maximize $\ln D(G(z))$), where $G(z)$ is the deterministic generator that maps latent variables z to the data space. Assuming $D = D^*$, this yields

$$\int_z p(z) \ln(1 - D^*(G(z)))\ \mathrm{d}z = \int_x p_\theta(x) \ln \frac{p_\theta(x)}{p_\theta(x) + p^*(x)}\ \mathrm{d}x = \mathcal{D}_{\text{KL}}(p_\theta||(p_\theta + p^*)/2), \tag{2}$$

which is a quality-driven criterion, favoring sample quality over support coverage.

4 Adaptive Density Estimation and hybrid adversarial-likelihood training

We present a hybrid training approach with MLE to cover the full support of the training data, and adversarial training as a trainable inductive bias mechanism to improve sample quality. Using both these criteria provides a richer training signal, but satisfying both criteria is more challenging than each in isolation for a given model complexity. In practice, model flexibility is limited by (i) the number of parameters, layers, and features in the model, and (ii) simplifying modeling assumptions, usually made for tractability. We show that these simplifying assumptions create a conflict between the two criteria, making successfull joint training non trivial. We introduce Adaptive Density Estimation as a solution to reconcile them.

Latent variable generative models, defined as $p_\theta(x) = \int_z p_\theta(x|z)p(z)\ \mathrm{d}z$, typically make simplifying assumptions on $p_\theta(x|z)$, such as full factorization and/or Gaussianity, see *e.g.* [11, 25, 30]. In particular, assuming full factorization of $p_\theta(x|z)$ implies that any correlations not captured by z are treated as independent per-pixel noise. This is a poor model for natural images, unless z captures each and every aspect of the image structure. Crucially, this hypothesis is problematic in the context of hybrid MLE-adversarial training. If p^* is too complex for $p_\theta(x|z)$ to fit it accurately enough, MLE will lead to a high variance in a factored (Gaussian) $p_\theta(x|z)$ as illustrated in Figure 4 (right).

This leads to unrealistic blurry samples, easily detected by an adversarial discriminator, which then does not provide a useful training signal. Conversely, adversarial training will try to avoid these poor samples by dropping modes of the training data, and driving the "noise" level to zero. This in turn is heavily penalized by maximum likelihood training, and leads to poor likelihoods on held-out data.

Adaptive density estimation. The point of view of regression hints at a possible solution. For instance, with isotropic Gaussian model densities with fixed variance, solving the optimization problem $\theta^* \in \max_\theta \ln(p_\theta(x|z))$ is similar to solving $\min_\theta ||\mu_\theta(z) - x||_2$, i.e., ℓ_2 regression, where $\mu_\theta(z)$ is the mean of the decoder $p_\theta(x|z)$. The Euclidean distance in RGB space is known to be a poor measure of similarity between images, non-robust to small translations or other basic transformations [34]. One can instead compute the Euclidean distance in a feature space, $||f_\psi(x_1) - f_\psi(x_2)||_2$, where f_ψ is chosen so that the distance is a better measure of similarity. A popular way to

obtain f_ψ is to use a CNN that learns a non-linear image representation, that allows linear assessment of image similarity. This is the idea underlying GAN discriminators, the FID evaluation measure [19], the reconstruction losses of VAE-GAN [27] and classifier based perceputal losses as in [21].

Despite their flexibility, such similarity metrics are in general degenerate in the sense that they may discard information about the data point x. For instance, two different images x and y can collapse to the same points in feature space, i.e., $f_\psi(x) = f_\psi(y)$. This limits the use of similarity metrics in the context of generative modeling for two reasons: (i) it does not yield a valid measure of likelihood over inputs, and (ii) points generated in the feature space f_ψ cannot easily be mapped to images. To resolve this issue, we chose f_ψ to be a bijection. Given a model p_θ trained to model $f_\psi(x)$ in feature space, a density in image space is computed using the change of variable formula, which yields $p_{\theta,\psi}(x) = p_\theta(f_\psi(x)) \left|\det \left(\partial f_\psi(x)/\partial x^\top\right)\right|$. Image samples are obtained by sampling from p_θ in feature space, and mapping to the image space through f_ψ^{-1}. We refer to this construction as Adaptive Denstiy Estimation. If p_θ provides efficient log-likelihood computations, the change of variable formula can be used to train f_ψ and p_θ together by maximum-likelihood, and if p_θ provides fast sampling adversarial training can be performed efficiently.

MLE with adaptive density estimation. To train a generative latent variable model $p_\theta(x)$ which permits efficient sampling, we rely on amortized variational inference. We use an inference network $q_\phi(z|x)$ to construct a variational evidence lower-bound (ELBO),

$$\mathcal{L}^\psi_{\text{ELBO}}(x, \theta, \phi) = \mathop{\mathbb{E}}_{q_\phi(z|x)} \left[\ln(p_\theta(f_\psi(x)|z))\right] - \mathcal{D}_{\text{KL}}(q_\phi(z|x)||p_\theta(z)) \leq \ln p_\theta(f_\psi(x)). \tag{3}$$

Using this lower bound together with the change of variable formula, the mapping to the similarity space f_ψ and the generative model p_θ can be trained jointly with the loss

$$\mathcal{L}_{\text{C}}(\theta, \phi, \psi) = \mathop{\mathbb{E}}_{x\sim p^*} \left[-\mathcal{L}^\psi_{\text{ELBO}}(x, \theta, \phi) - \ln \left|\det \frac{\partial f_\psi(x)}{\partial x^\top}\right|\right] \geq - \mathop{\mathbb{E}}_{x\sim p^*} \left[\ln p_{\theta,\psi}(x)\right]. \tag{4}$$

We use gradient descent to train f_ψ by optimizing $\mathcal{L}_{\text{C}}(\theta, \phi, \psi)$ w.r.t. ψ. The $\mathcal{L}_{\text{ELBO}}$ term encourges the mapping f_ψ to maximize the density of points in feature space under the model p_θ, so that f_ψ is trained to match modeling assumptions made in p_θ. Simultaneously, the log-determinant term encourages f_ψ to maximize the volume of data points in feature space. This guarantees that data points cannot be collapsed to a single point in the feature space. We use a factored Gaussian form of the conditional $p_\theta(.|z)$ for tractability, but since f_ψ can arbitrarily reshape the corresponding conditional image space, it still avoids simplifying assumptions in the image space. Therefore, the (invertible) transformation f_ψ avoids the conflict between the MLE and adversarial training mechanisms, and can leverage both.

Adversarial training with adaptive density estimation. To sample the generative model, we sample latents from the prior, $z \sim p_\theta(z)$, which are then mapped to feature space through $\mu_\theta(z)$, and to image space through f_ψ^{-1}. We train our generator using the modified objective proposed by [50], combining both generator losses considered in [15], *i.e.* $\ln[(1 - D(G(z)))/D(G(z))]$, which yields:

$$\mathcal{L}_{\text{Q}}(p_{\theta,\psi}) = - \mathop{\mathbb{E}}_{p_\theta(z)} \left[\ln D(f_\psi^{-1}(\mu_\theta(z))) - \ln(1 - D(f_\psi^{-1}(\mu_\theta(z))))\right]. \tag{5}$$

Assuming the discriminator D is trained to optimality at every step, it is easy to demonstrate that the generator is trained to optimize $\mathcal{D}_{\text{KL}}(p_{\theta,\psi}||p^*)$. The training procedure, written as an algorithm in Appendix H, alternates between (i) bringing $\mathcal{L}_{\text{Q}}(p_{\theta,\psi})$ closer to it's optimal value $\mathcal{L}^*_{\text{Q}}(p_{\theta,\psi}) = \mathcal{D}_{\text{KL}}(p_{\theta,\psi}||p^*)$, and (ii) minimizing $\mathcal{L}_{\text{C}}(p_{\theta,\psi}) + \mathcal{L}_{\text{Q}}(p_{\theta,\psi})$. Assuming that the discriminator is trained to optimality at every step, the generative model is trained to minimize a bound on the symmetric sum of two KL divergences: $\mathcal{L}_{\text{C}}(p_{\theta,\psi}) + \mathcal{L}^*_{\text{Q}}(p_{\theta,\psi}) \geq \mathcal{D}_{\text{KL}}(p^*||p_{\theta,\psi}) + \mathcal{D}_{\text{KL}}(p_{\theta,\psi}||p^*) + \mathcal{H}(p^*)$, where the entropy of the data generating distribution, $\mathcal{H}(p^*)$, is an additive constant independent of the generative model $p_{\theta,\psi}$. In contrast, MLE and GANs optimize one of these divergences each.

5 Experimental evaluation

We present our evaluation protocol, followed by an ablation study to assess the importance of the components of our model (Section 5.1). We then show the quantitative and qualitative performance of our model, and compare it to the state of the art on the CIFAR-10 dataset in Section 5.2. We present additional results and comparisons on higher resolution datasets in Section 5.3.

Figure 5: Samples from GAN and VAE baselines, our V-ADE, AV-GDE and AV-ADE models, all trained on CIFAR-10.

	f_ψ	Adv.	MLE	BPD ↓	IS ↑	FID ↓
GAN	×	✓	×	[7.0]	6.8	31.4
VAE	×	×	✓	4.4	2.0	171.0
V-ADE†	✓	×	✓	3.5	3.0	112.0
AV-GDE	×	✓	✓	4.4	5.1	58.6
AV-ADE†	✓	✓	✓	3.9	7.1	28.0

Table 1: Quantitative results. † : Parameter count decreased by 1.4% to compensate for f_ψ. [Square brackets] denote that the value is approximated, see Section 5.

Evalutation metrics. We evaluate our models with complementary metrics. To assess sample quality, we report the Fréchet inception distance (FID) [19] and the inception score (IS) [45], which are the de facto standard metrics to evaluate GANs [5, 54]. Although these metrics focus on sample quality, they are also sensitive to coverage, see Appendix D for details. To specifically evaluate the coverage of held-out data, we use the standard bits per dimension (BPD) metric, defined as the negative log-likelihood on held-out data, averaged across pixels and color channels [9].

Due to their degenerate low-dimensional support, GANs do not define a density in the image space, which prevents measuring BPD on them. To endow a GAN with a full support and a likelihood, we train an inference network "around it", while keeping the weights of the GAN generator fixed. We also train an isotropic noise parameter σ. For both GANs and VAEs, we use this inference network to compute a lower bound to approximate the likelihood, *i.e.*, an upper bound on BPD. We evaluate all metrics using held-out data not used during training, which improves over common practice in the GAN literature, where training data is often used for evaluation.

5.1 Ablation study and comparison to VAE and GAN baselines

We conduct an ablation study on the CIFAR-10 dataset.[1] Our GAN baseline uses the non-residual architecture of SNGAN [37], which is stable and quick to train, without spectral normalization. The same convolutional architecture is kept to build a VAE baseline.[2] It produces the mean of a factorizing Gaussian distribution. To ensure a valid density model we add a trainable isotropic variance σ. We train the generator for coverage by optimizing $\mathcal{L}_Q(p_\theta)$, for quality by optimizing $\mathcal{L}_C(p_\theta)$, and for both by optimizing the sum $\mathcal{L}_Q(p_\theta) + \mathcal{L}_C(p_\theta)$. The model using Variational inference with Adaptive Density Estimation (ADE) is refered to as V-ADE. The addition of adversarial training is denoted AV-ADE, and hybrid training with a Gaussian decoder as AV-GDE. The bijective function f_ψ, implemented as a small Real-NVP with 1 scale, 3 residual blocks, 2 layers per block, increases the number of weights by approximately 1.4%. We compensate for these additional parameters with a slight decrease in the width of the generator for fair comparison.[3] See Appendix B for details.

Experimental results in Table 1 confirm that the GAN baseline yields better sample quality (IS and FID) than the VAE baseline: obtaining inception scores of 6.8 and 2.0, respectively. Conversely, VAE achieves better coverage, with a BPD of 4.4, compared to 7.0 for GAN. An identical generator trained for both quality and coverage, AV-GDE, obtains a sample quality that is in between that of the GAN and the VAE baselines, in line with the analysis in Section 4. Samples from the different models in Figure 5 confirm these quantitative results. Using f_ψ and training with $\mathcal{L}_C(p_\theta)$ only, denoted by V-ADE in the table, leads to improved sample quality with IS up from 2.0 to 3.0 and FID down from 171 to 112. Note that this quality is still below the GAN baseline and our AV-GDE model.

When f_ψ is used with coverage and quality driven training, AV-ADE, we obtain improved IS and FID scores over the GAN baseline, with IS up from 6.8 to 7.1, and FID down from 31.4 to 28.0. The

[1] We use the standard split of 50k/10k train/test images of 32×32 pixels. [2] In the VAE model, some intermediate feature maps are treated as conditional latent variables, allowing for hierarchical top-down sampling (see Appendix B). Experimentally, we find that similar top-down sampling is not effective for the GAN model.
[3] This is, however, too small to have a significant impact on the experimental results.

examples shown in the figure confirm the high quality of the samples generated by our AV-ADE model. Our model also achieves a better BPD than the VAE baseline. These experiments demonstrate that our proposed bijective feature space substantially improves the compatibility of coverage and quality driven training. We obtain improvements over both VAE and GAN in terms of held-out likelihood, and improve VAE sample quality to, or beyond, that of GAN. We further evaluate our model using the recent precision and recall approach of [43] an the classification framework of [48] in Appendix E. Additional results showing the impact of the number of layers and scales in the bijective similarity mapping f_ψ (Appendix F), reconstructions qualitatively demonstrating the inference abilities of our AV-ADE model (Appendix G) are presented in the supplementary material.

5.2 Refinements and comparison to the state of the art

We now consider further refinements to our model, inspired by recent generative modeling literature. Four refinements are used: (i) adding residual connections to the discriminator [17] (rd), (ii) leveraging more accurate posterior approximations using inverse auto-regressive flow [26] (iaf); see Appendix B, (iii) training wider generators with twice as many channels (wg), and (iv) using a hierarchy of two scales to build f_ψ (s2); see Appendix F. Table 2 shows consistent improvements with these additions, in terms of BPD, IS, FID.

Refinements	BPD ↓	IS ↑	FID ↓
GAN	[7.0]	6.8	31.4
GAN (rd)	[6.9]	7.4	24.0
AV-ADE	3.9	7.1	28.0
AV-ADE (rd)	3.8	7.5	26.0
AV-ADE (wg, rd)	3.8	**8.2**	**17.2**
AV-ADE (iaf, rd)	3.7	8.1	18.6
AV-ADE (s2)	**3.5**	6.9	28.9

Table 2: Model refinements.

Table 3 compares our model to existing hybrid approaches and state-of-the-art generative models on CIFAR-10. In the category of hybrid models that define a valid likelihood over the data space, denoted by Hybrid (L) in the table, FlowGAN(H) optimizes MLE and an adversarial loss, and FlowGAN(A) is trained adversarially. The AV-ADE model significantly outperforms these two variants both in terms of BPD, from 4.2 to between 3.5 and 3.8, and quality, *e.g.*, IS improves from 5.8 to 8.2. Compared to models that train an inference network adversarially, denoted by Hybrid (A), our model shows a substantial improvement in IS from 7.0 to 8.2. Note that these models do not allow likelihood evaluation, thus BPD values are not defined.

Compared to adversarial models, which are not optimized for support coverage, AV-ADE obtains better FID (17.2 down from 21.7) and similar IS (8.2 for both) compared to SNGAN with residual connections and hinge-loss, despite training on 17% less data than GANs (test split removed). The improvement in FID is likely due to this measure being more sensitive to support coverage than IS. Compared to models optimized with MLE only, we obtain a BPD between 3.5 and 3.7, comparable to 3.5 for Real-NVP demonstrating a good coverage of the support of held-out data. We computed IS and FID scores for MLE based models using publicly released code, with provided parameters (denoted by † in the table) or trained ourselves (denoted by ‡). Despite being smaller (for reference Glow has 384 layers *vs*. at most 10 for our deeper generator), our AV-ADE model generates better samples, *e.g.*, IS up from 5.5 to 8.2 (samples displayed in Figure 6), owing to quality driven training controling where the off-dataset mass goes. Additional samples from our AV-ADE model and comparison to others models are given in Appendix A.

Hybrid (L)	BPD ↓	IS ↑	FID ↓
AV-ADE (wg, rd)	3.8	**8.2**	**17.2**
AV-ADE (iaf, rd)	3.7	8.1	18.6
AV-ADE (S2)	**3.5**	6.9	28.9
FlowGan(A) [16]	8.5	5.8	
FlowGan(H) [16]	4.2	3.9	

Hybrid (A)	BPD ↓	IS ↑	FID ↓
AGE [52]		5.9	
ALI [12]		5.3	
SVAE [6]		6.8	
α-GAN [41]		6.8	
SVAE-r [6]		**7.0**	

Adversarial	BPD ↓	IS ↑	FID ↓
mmd-GAN[1]		7.3	25.0
SNGan [37]		7.4	29.3
BatchGAN [32]		7.5	23.7
WGAN-GP [17]		7.9	
SNGAN$_{(R,H)}$		**8.2**	**21.7**

MLE	BPD ↓	IS ↑	FID ↓
Real-NVP [9]	3.5	4.5†	56.8†
VAE-IAF [26]	3.1	3.8†	73.5†
Pixcnn++ [46]	**2.9**	5.5	
Flow++ [20]	3.1		
Glow [24]	3.4	**5.5**‡	**46.8**‡

Table 3: Performance on CIFAR10, without labels. MLE and Hybrid (L) models discard the test split. †: computed by us using provided weights. ‡: computed by us using provided code to (re)train models.

5.3 Results on additional datasets

To further validate our model we evaluate it on STL10 (48×48), ImageNet and LSUN (both 64×64). We use a wide generator to account for the higher resolution, without IAF, a single scale in f_ψ, and no residual blocks (see Section 5.2). The architecture and training hyper-parameters are not tuned, besides adding one layer at resolution 64×64, which demonstrates the stability of our approach. On STL10, Table 4 shows that our AV-ADE improves inception score over SNGAN, from 9.1 up to

Glow @ 3.35 BPD | FlowGan (H) @ 4.21 BPD | AV-ADE (iaf, rd) @ 3.7 BPD

Figure 6: Samples from models trained on CIFAR-10. Our AV-ADE spills less mass on unrealistic samples, owing to adversarial training which controls where off-dataset mass goes.

9.4, and is second best in FID. Our likelihood performance, between 3.8 and 4.4, and close to that of Real-NVP at 3.7, demonstrates good coverage of the support of held-out data. On the ImageNet dataset, maintaining high sample quality, while covering the full support is challenging, due to its very diverse support. Our AV-ADE model obtains a sample quality behind that of MMD-GAN with IS/FID scores at $8.5/45.5$ *vs*. $10.9/36.6$. However, MMD-GAN is trained purely adversarially and does not provide a valid density across the data space, unlike our approach.

Figure 7 shows samples from our generator trained on a single GPU with 11 Gb of memory on LSUN classes. Our model yields more compelling samples compared to those of Glow, despite having less layers (7 *vs*. over 500). Additional samples on other LSUN categories are presented in Appendix A.

STL-10 (48×48)	BPD ↓	IS ↑	FID ↓
AV-ADE (wg, wd)	4.4	**9.4**	44.3
AV-ADE (iaf, wd)	4.0	8.6	52.7
AV-ADE (s2)	3.8	8.6	52.1
Real-NVP	**3.7**‡	4.8‡	103.2‡
BatchGAN		8.7	51
SNGAN (Res-Hinge)		9.1	**40.1**

ImageNet (64×64)	BPD ↓	IS ↑	FID ↓
AV-ADE (wg, wd)	4.90	8.5	45.5
Real-NVP	3.98		
Glow	3.81		
Flow++	**3.69**		
MMD-GAN		**10.9**	**36.6**

LSUN	Real-NVP	Glow	Ours
Bedroom	(2.72/×)	(2.38/208.8†)	(3.91, **21.1**)
Tower	(2.81/×)	(2.46/214.5†)	(3.95, **15.5**)
Church	(3.08/×)	(2.67/222.3†)	(4.3, **13.1**)
Classroom	×	×	(4.6, 20.0)
Restaurant	×	×	(4.7, 20.5)

Table 4: Results on the STL-10, ImageNet, and LSUN datasets. AV-ADE (wg, rd) is used for LSUN.

Figure 7: Samples from models trained on LSUN Churches (C) and bedrooms (B). Our AV-ADE model over-generalises less and produces more compelling samples. See Appendix A for more classes and samples.

6 Conclusion

We presented a generative model that leverages invertible network layers to relax the conditional independence assumptions commonly made in VAE decoders. It allows for efficient feed-forward sampling, and can be trained using a maximum likelihood criterion that ensures coverage of the data generating distribution, as well as an adversarial criterion that ensures high sample quality.

Acknowledgments. The authors would like to thank Corentin Tallec, Mathilde Caron, Adria Ruiz and Nikita Dvornik for useful feedback and discussions. Acknowledgments also go to our anonymous reviewers, who contributed valuable comments and remarks.
This work was supported in part by the grants ANR16-CE23-0006 "Deep in France", LabEx PERSYVAL-Lab (ANR-11-LABX0025-01) as well as the Indo-French project EVEREST (no. 5302-1) funded by CEFIPRA and a grant from ANR (AVENUE project ANR-18-CE23-0011),

References

[1] M. Arbel, D. J. Sutherland, M. Binkowski, and A. Gretton. On gradient regularizers for MMD GANs. In *NeurIPS*, 2018.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017.

[3] P. Bachman. An architecture for deep, hierarchical generative models. In *NeurIPS*, 2016.

[4] C. Bishop. *Pattern recognition and machine learning*. Springer-Verlag, 2006.

[5] A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.

[6] L. Chen, S. Dai, Y. Pu, E. Zhou, C. Li, Q. Su, C. Chen, and L. Carin. Symmetric variational autoencoder and connections to adversarial learning. In *AISTATS*, 2018.

[7] X. Chen, D. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. Variational lossy autoencoder. In *ICLR*, 2017.

[8] H. De Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. Courville. Modulating early visual processing by language. In *NeurIPS*, 2017.

[9] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using Real NVP. In *ICLR*, 2017.

[10] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. In *ICLR*, 2017.

[11] G. Dorta, S. Vicente, L. Agapito, N. Campbell, and I. Simpson. Structured uncertainty prediction networks. In *CVPR*, 2018.

[12] V. Dumoulin, I. Belghazi, B. Poole, A. Lamb, M. Arjovsky, O. Mastropietro, and A. Courville. Adversarially learned inference. In *ICLR*, 2017.

[13] V. Dumoulin, J. Shlens, and M. Kudlur. A learned representation for artistic style. In *ICLR*, 2017.

[14] M. Elfeki, C. Couprie, M. Riviere, and M. Elhoseiny. GDPP: Learning diverse generations using determinantal point processes. In *arxiv.org/pdf/1812.00068*, 2018.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[16] A. Grover, M. Dhar, and S. Ermon. Flow-GAN: Combining maximum likelihood and adversarial learning in generative models. In *AAAI Press*, 2018.

[17] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of Wasserstein GANs. In *NeurIPS*, 2017.

[18] I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taiga, F. Visin, D. Vazquez, and A. Courville. PixelVAE: A latent variable model for natural images. In *ICLR*, 2017.

[19] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *NeurIPS*, 2017.

[20] J. Ho, X. Chen, A. Srinivas, Y. Duan, and P. Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *ICML*, 2019.

[21] X. Hou, L. Shen, K. Sun, and G. Qiu. Deep feature consistent variational autoencoder. In *WACV*, 2017.

[22] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018.

[23] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[24] D. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*, 2018.

[25] D. Kingma and M. Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.

[26] D. Kingma, T. Salimans, R. Józefowicz, X. Chen, I. Sutskever, and M. Welling. Improving variational autoencoders with inverse autoregressive flow. In *NeurIPS*, 2016.

[27] A. Larsen, S. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In *ICML*, 2016.

[28] Y. Li, K. Swersky, and R. Zemel. Generative moment matching networks. In *ICML*, 2015.

[29] Z. Lin, A. Khetan, G. Fanti, and S. Oh. PacGAN: The power of two samples in generative adversarial networks. In *NeurIPS*, 2018.

[30] O. Litany, A. Bronstein, M. Bronstein, and A. Makadia. Deformable shape completion with graph convolutional autoencoders. In *CVPR*, 2018.

[31] T. Lucas and J. Verbeek. Auxiliary guided autoregressive variational autoencoders. In *ECML*, 2018.

[32] T. Lucas, C. Tallec, Y. Ollivier, and J. Verbeek. Mixed batches and symmetric discriminators for GAN training. In *ICML*, 2018.

[33] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow. Adversarial autoencoders. In *ICLR*, 2016.

[34] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016.

[35] J. Menick and N. Kalchbrenner. Generating high fidelity images with subscale pixel networks and multidimensional upscaling. In *ICLR*, 2019.

[36] T. Miyato and M. Koyama. cGANs with projection discriminator. In *ICLR*, 2018.

[37] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

[38] P. Ramachandran, T. Paine, P. Khorrami, M. Babaeizadeh, S. Chang, Y. Zhang, M. Hasegawa-Johnson, R. Campbell, and T. Huang. Fast generation for convolutional autoregressive models. In *ICLR workshop*, 2017.

[39] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *ICML*, 2015.

[40] D. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

[41] M. Rosca, B. Lakshminarayanan, D. Warde-Farley, and S. Mohamed. Variational approaches for auto-encoding generative adversarial networks. *arXiv preprint arXiv:1706.04987*, 2017.

[42] Y. Saatchi and A. Wilson. Bayesian GAN. In *NeurIPS*, 2017.

[43] M. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly. Assessing generative models via precision and recall. In *NeurIPS*, 2018.

[44] T. Salimans and D. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *NeurIPS*, 2016.

[45] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training GANs. In *NeurIPS*, 2016.

[46] T. Salimans, A. Karpathy, X. Chen, and D. Kingma. PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications. In *ICLR*, 2017.

[47] T. Salimans, H. Zhang, A. Radford, and D. Metaxas. Improving gans using optimal transport. In *ICLR*, 2018.

[48] K. Shmelkov, C. Schmid, and K. Alahari. How good is my GAN? In *ECCV*, 2018.

[49] C. Sønderby, T. Raiko, L. Maaløe, S. Sønderby, and O. Winther. Ladder variational autoencoders. In *NeurIPS*, 2016.

[50] C. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár. Amortised MAP inference for image super-resolution. In *ICLR*, 2017.

[51] H. Thanh-Tung, T. Tran, and S. Venkatesh. Improving generalization and stability of generative adversarial networks. In *ICLR*, 2019.

[52] D. Ulyanov, A. Vedaldi, and V. Lempitsky. It takes (only) two: Adversarial generator-encoder networks. In *AAAI*, 2018.

[53] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016.

[54] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. In *ICML*, 2019.

A Quantitative and qualitative results

In this section we provide additional quantitative and qualitative results on CIFAR10, STL10, LSUN (categories: Bedrooms, Towers, Bridges, Kitchen, Church, Living room, Dining room, Classroom, Conference room and Restaurant) at resolutions 64×64 and 128×128, ImageNet and CelebA. We report IS/FID scores together with BPD.

A.1 Additional samples on CIFAR10

Samples from AV-ADE (wg, rd) Real images

Figure 8: Samples from our AV-ADE (wg, rd) model trained on CIFAR10 compared to real images. A significant proportion of samples can be reasonably attributed to a CIFAR10 class, even though the model was trained without labels. Our model is constrained to cover the full support of the data, which translates to diverse samples, as noted in the qualitative results illustrated here.

A.2 Quantitative results on LSUN categories

LSUN (64×64)	BPD↓	FID↓
Bedrooms	3.92	21.1
Towers	3.95	15.5
Bridges	4.1	25.2
Kitchen	4.0	15.3
Church	4.3	13.2
Living	4.1	16.7
Dining	4.3	14.6
Classroom	4.6	20.0
Conf. room	4.2	21.3
Restaurant	4.7	20.5

LSUN (128×128)	BPD↓	FID↓
Bedrooms	3.0	70.1
Towers	3.1	40.8
Bridges	3.4	64.2
Kitchen	3.1	45.2
Church	3.5	47.3
Living	3.2	50.6
Dining	3.2	41.0
Classroom	3.5	59.2
Conf. room	3.1	61.7
Restaurant	3.8	60.0

Table 5: Quantitative results obtained on LSUN categories, at resolutions 64×64 and 128×128 using our AV-ADE (wg, rd) model.

A.3 Samples on all Lsun datasets

Figure 9: Samples obtained from our AV-ADE (wg, rd) model trained on LSUN bedrooms (left), compared to training images (right).

Figure 10: Samples obtained from our AV-ADE (wg, rd) model trained on LSUN towers (left), compared to training images (right).

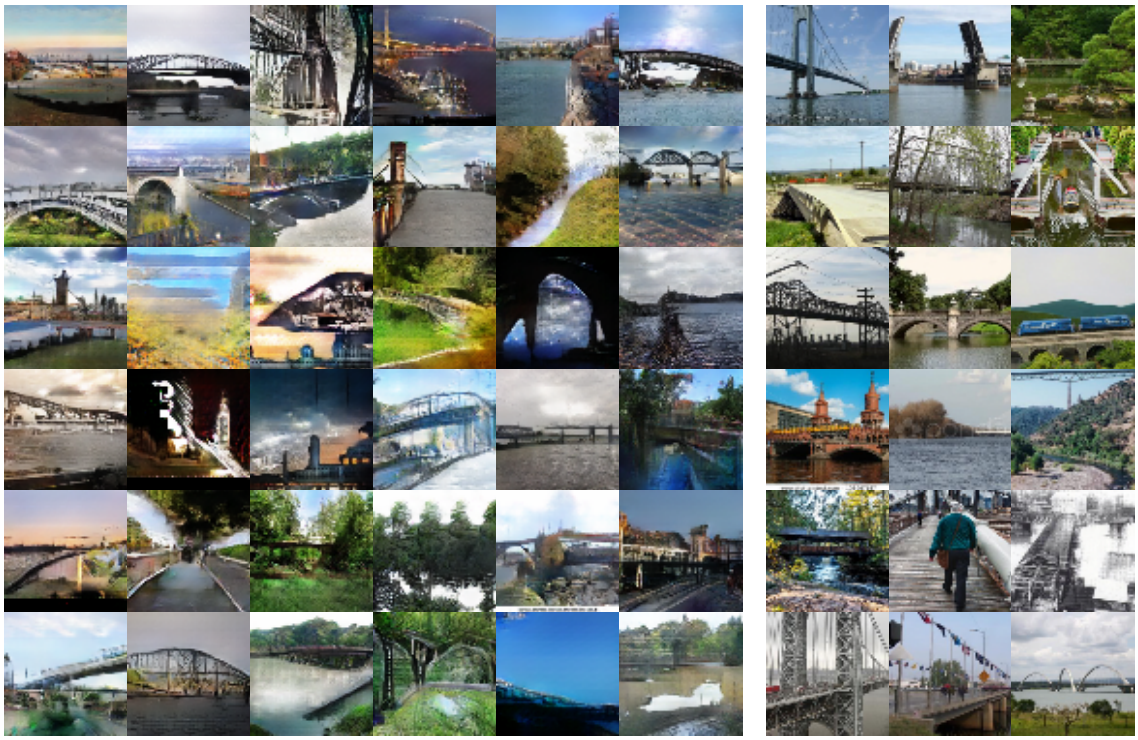


Figure 11: Samples obtained from our AV-ADE (wg, rd) model trained on LSUN bridges (left), compared to training images (right).

Figure 12: Samples obtained from our AV-ADE (wg, rd) model trained on LSUN kitchen (left), compared to training images (right).

Figure 13: Samples obtained from our AV-ADE (wg, rd) model trained on LSUN churches (left), compared to training images (right).

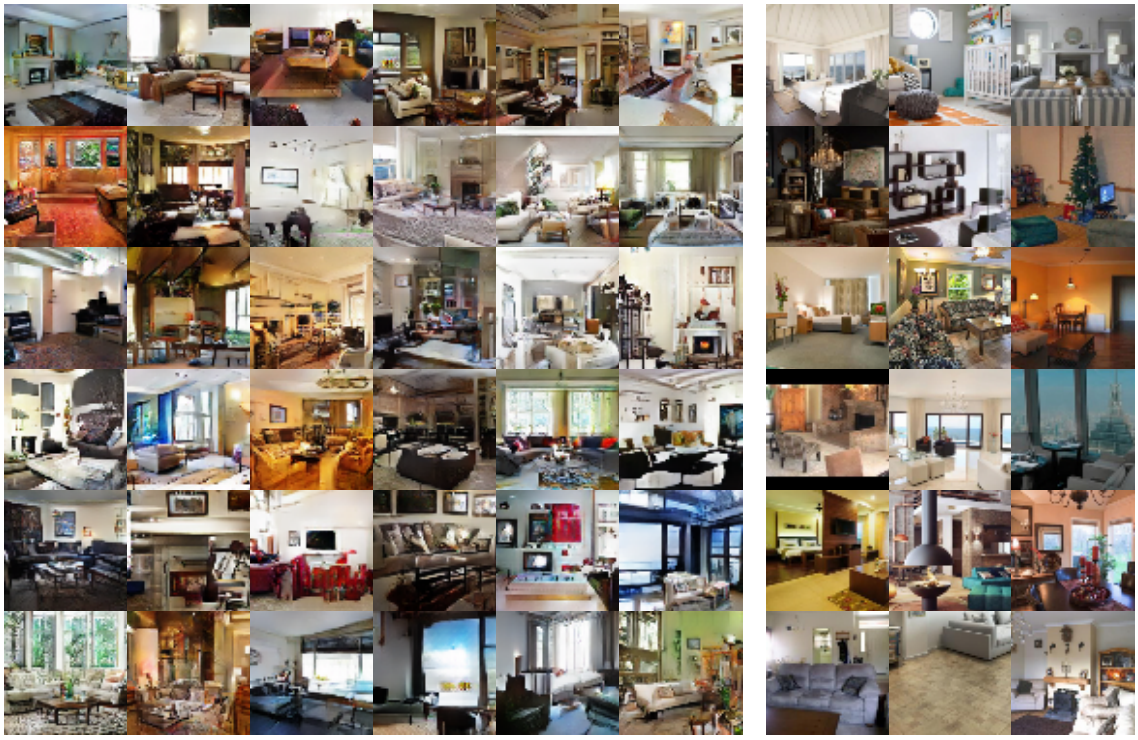


Figure 14: Samples obtained from our AV-ADE (wg, rd) model trained on LSUN living rooms (left), compared to training images (right).

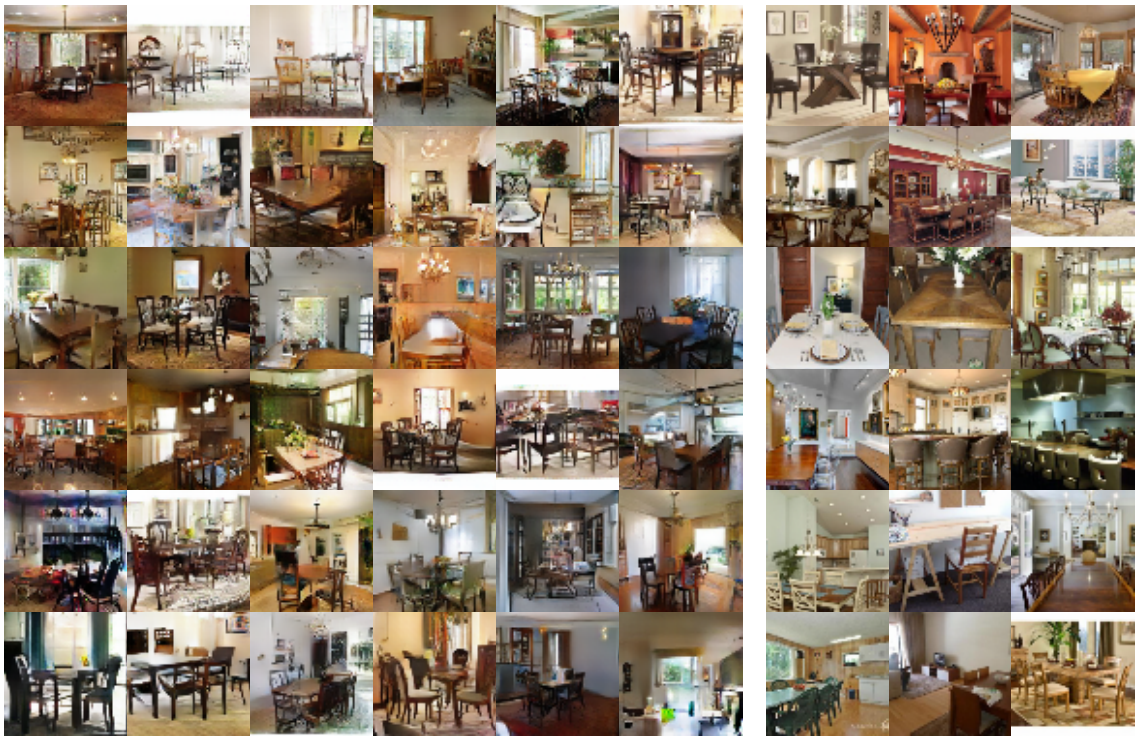
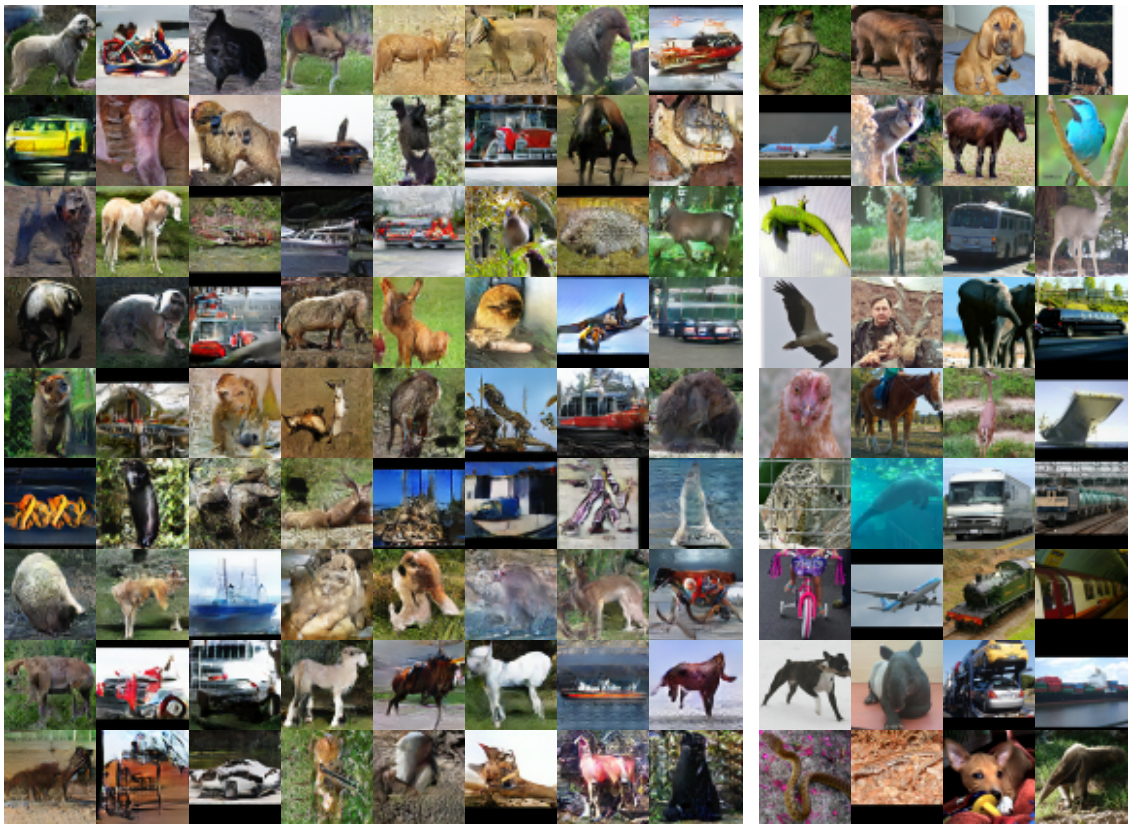


Figure 15: Samples obtained from our AV-ADE (wg, rd) model trained on LSUN dining rooms (left), compared to training images (right).

Figure 16: Samples obtained from our AV-ADE (wg, rd) model trained on LSUN conference rooms (left), compared to training images (right).

Figure 17: Samples obtained from our AV-ADE (wg, rd) model trained on LSUN restaurants (left), compared to training images (right).

A.4 Additional samples on STL10



Samples from AV-ADE (wg, rd) Real images

Figure 18: Samples from our AV-ADE (wg, rd) model trained on STL10 compared to real images. Training was done without any label conditioning.

B Model refinements

B.1 Top-down sampling of hierarchical latent variables

Flexible priors and posteriors for the variational autoencoder model can be obtained by sampling hierarchical latent variables at different layers in the network. In the generative model p_θ, latent variables $\mathbf{z}$ can be split into L groups, each one at a different layer, and the density over $\mathbf{z}$ can be written as:

$$q(\mathbf{z}) = q(\mathbf{z}_L) \prod_{i=1}^{L-1} q(\mathbf{z}_i|\mathbf{z}_{i+1}). \tag{6}$$

Additionally, to allow the chain of latent variables to be sampled in the same order when encoding-decoding and when sampling, top-down sampling is used, as proposed in Bachman [3], Kingma et al. [26], Sønderby et al. [49]. With top-down sampling, the encoder (symmetric to the decoder) extracts deterministic features h_i at different levels as the image is being encoded, constituting the bottom-up deterministic pass. While decoding the image, these previously extracted deterministic features h_i are used for top-down sampling and help determining the posterior over latent variables at different depths in the decoder. These posteriors are also conditioned on the latent variables sampled at lower feature resolutions, using normal densities as follows:

$$q_\phi(\mathbf{z}_1|x) = \mathcal{N}(\mathbf{z}_1|\mu_1(x, h_1), \sigma_1^2(x, h_1)), \tag{7}$$

$$q_\phi(\mathbf{z}_i|\mathbf{z}_{i-1}) = \mathcal{N}(\mathbf{z}_i|\mu_i(x, \mathbf{z}_{i-1}, h_{i-1}), \sigma_i^2(x, \mathbf{z}_{i-1}, h_{i-1})). \tag{8}$$

This constitutes the stochastic top-down pass. We refer the reader to Bachman [3], Kingma et al. [26], Sønderby et al. [49] for more detail.

B.2 Inverse autoregressive flow

To increase the flexibility of posteriors used over latent variables in variational inference, Kingma et al. [26] proposed a type of normalizing flow called inverse autoregressive flow (IAF). The main benefits of this normalizing flow are its scalability to high dimensions, and its ability to leverage autoregressive neural network, such as those introduced by van den Oord et al. [53]. First, a latent variable vector is sampled using the reparametrization trick [25]:

$$\epsilon \sim \mathcal{N}(0, I), z_0 = \mu_0 + \sigma_0 \epsilon. \tag{9}$$

Then, mean and variance parameters μ_1 and σ_1 are computed as functions of z_0 using autoregressive models, and a new latent variable z_1 is obtained:

$$z_1 = \mu_1(z_0) + \sigma_1(z_0) z_0. \tag{10}$$

Since σ_1 and μ_1 are implemented by autoregressive networks, the Jacobian $\frac{dz_1}{dz_0}$ is triangular with the values of σ_1 on the diagonal, and the density under the new latent variable remains efficient to compute. This transformation can be repeated an arbitrary number of times for increased flexibility in theory, but in practice a single step is used.

B.3 Gradient penalty

A body of work on generative adversarial networks centers around the idea of regularizing the discriminator by enforcing Lipschitz continuity, for instance by Arjovsky et al. [2], Gulrajani et al. [17], Miyato et al. [37], Thanh-Tung et al. [51]. In this work, we use the approach of Gulrajani et al. [17], that enforces the Lipschitz constraint with a gradient penalty term added to the loss:

$$\mathcal{L}_{Grad} = \lambda + \mathbb{E}_{\hat{x}}[(||\Delta_{\hat{x}} D(\hat{x})||_2 - 1)^2], \tag{11}$$

where $\hat{x}$ is obtained by interpolating between real and generated data:

$$\epsilon \sim U_{[0,1]}, \tag{12}$$

$$\hat{x} = \epsilon x + (1 - \epsilon)\tilde{x}. \tag{13}$$

We add this term to the loss used to train the discriminator that yields our quality driven criterion.

C Implementation details

Architecture and training hyper-parameters. We used Adamax [23] with learning rate 0.002, $\beta_1 = 0.9$, $\beta_2 = 0.999$ for all experiments. The convolutional architecture of the models used is described in Table 6a and Table 6b. All CIFAR-10 experiments use batch size 64, other experiments in high resolution use batch size 32. To stabilize the adversarial training we use the gradient penalty [17] with coefficient 100, and 1 discriminator update per generator update. We experimented with different weighting coefficients between the two loss components, and found that values in the range 10 to 100 on the adversarial component work best in practice. No significant influence on the final performance of the model is observed in this range, though the training dynamics in early training are improved with higher values. With values significantly smaller than 10, discriminator collapse was observed in a few isolated cases. All experiments reported here use coefficient 100.

For experiments with hierarchical latent variables, we use 32 of them per layer. In the generator we use ELU nonlinearity, in discriminator with residual blocks we use ReLU, while in simple convolutional discriminator we use leaky ReLU with slope 0.2.

Unless stated otherwise we use three Real-NVP layers with a single scale and two residual blocks that we train only with the likelihood loss. Regardless of the number of scales, the VAE decoder always outputs a tensor of the same dimension as the target image, which is then fed to the Real-NVP layers. As in the reference implementations, we use both batch normalization and weight normalization in Real-NVP and only weight normalization in IAF. We use the reference implementations of IAF and Real-NVP released by the authors.

Discriminator
conv 3×3, 16
ResBlock 32
ResBlock down 64
ResBlock down 128
ResBlock down 256
Average pooling
dense 1

Generator
conv 3×3, 16
IAF block 32
IAF block down 64
IAF block down 128
IAF block down 256
$h \sim \mathcal{N}(0; 1)$
IAF block up 256
IAF block up 128
IAF block up 64
IAF block 32
conv 3×3, 3

Table 6: Residual architectures for experiments from Section 5.2 and Table 9

D On the Inception Score and the Fréchet inception distance

Quantitative evaluation of Generative Adversarial Networks is challenging, in part due to the absence of log-likelihood. Inception score (IS) and Fréchet Inception distance (FID) are two measures proposed by [45] and [19] respectively, to automate the qualitative evaluation of samples. Though imperfect, they have been shown to correlate well with human judgement in practice, and it is standard in GAN literature to use them for quantitative evaluations. These metrics are also sensitive to coverage. Indeed, any metric evaluating quality only would be degenerate, as collapsing to the mode of the distribution would maximize it. However, in practice both metrics are much more sensitive to quality than to support coverage, as we evaluate below.

Inception score (IS) [45] is a statistic of the generated images, based on an external deep network trained for classification on ImageNet. It is given by:

$$IS(p_\theta) = \exp\left(\mathbb{E}_{x\sim p_\theta} \mathcal{D}_{\mathrm{KL}}\left(p\left(y|x\right)||p(y)\right)\right), \tag{14}$$

where $x \sim p_\theta$ is sampled from the generative model, $p(y|x)$ is the conditional class distribution obtained by applying the pretrained classification network to the generated images, and $p(y) = \int_x p(y|x)p_\theta(x)$ is the class marginal over the generated images.

Fréchet Inception distance (FID) [19] compares the distributions of Inception embeddings *i.e.*, activations from the penultimate layer of the Inception network, of real ($p_r(\mathbf{x})$) and generated ($p_g(\mathbf{x})$) images. Both these distributions are modeled as multi-dimensional Gaussians parameterized by their respective mean and covariance. The distance measure is defined between the two Gaussian distributions as:

$$d^2((\mathbf{m}_r, \mathbf{C}_r), (\mathbf{m}_g, \mathbf{C}_g)) = \|\mathbf{m}_r - \mathbf{m}_g\|^2 + \mathrm{Tr}(\mathbf{C}_r + \mathbf{C}_g - 2(\mathbf{C}_r\mathbf{C}_g)^{\frac{1}{2}}), \tag{15}$$

where $(\mathbf{m}_r, \mathbf{C}_r)$, $(\mathbf{m}_g, \mathbf{C}_g)$ denote the mean and covariance of the real and generated image distributions respectively.

Practical use. IS and FID correlate predominantly with the quality of samples. In GAN literature, for instance Miyato et al. [37], they are considered to correlate well with human judgement of quality. An empirical indicator of that is that state-of-the art likelihood-based models have very low IS/FID scores despite having good coverage, which shows that the low quality of their samples impacts IS/FID more heavily than their coverage performance. Conversely, state-of-the art adversarial models have high IS/FID scores, despite suffering from mode dropping (which strongly degrades BPD). So the score is determined mainly by the high quality of their samples. This is especially true when identical architectures and training budget are considered, as in our first experiment in Section 5.1.

To obtain a quantitative estimation of how much entropy/coverage impacts the IS and FID measures, we evaluate the scores obtained by random subsamples of the dataset, such that the quality is unchanged but coverage progressively degrades (see details of the scores below). Table 7 shows that when using the full set of images (50k) the FID is 0 as the distributions are identical. Notice that as the number of images decreases, IS is very stable (it can even increase, but by very low increments that fall below statistical noise, with a typical standard deviation of 0.1). This is because the entropy of the

Split size	IS	FID
50k (full)	11.3411	0.00
40k	11.3388	0.13
30k	11.3515	0.35
20k	11.3458	0.79
10k	11.3219	2.10
5k	11.2108	4.82
2.5k	11.0446	10.48

Table 7: IS and FID scores obtained by the ground truth when progressively dropping parts of the dataset. The metrics are largely insensitive to removing most of the dataset, unlike BPD. For reference, a reasonable GAN could get around 8 IS and 20 FID.

distribution is not strongly impacted by subsampling, even though coverage is. FID is more sensitive, as it behaves more like a measure of coverage (it compares the two distributions). Nonetheless, the variations remain extremely low even when dropping most of the dataset. For instance, when removing 80% of the dataset (*i.e.*, using 10k images), FID is at 2.10, to be compared with typical GAN/AV-ADE values that are around 20. These measurements demonstrate that IS and FID scores are heavily dominated by the quality of images. From this, we conclude that IS and FID can be used as reasonable proxies to asses sample quality, even though they are also slightly influenced by coverage. One should bear in mind, however, that a small increase in these scores may come from better coverage rather than improved sample quality.

E Other evaluation metrics

E.1 Evaluation using samples as training data for a discriminator

We also evaluate our approach using the two metrics recently proposed by Shmelkov et al. [48]. This requires a class-conditional version of AV-ADE. To address the poor compatibility of conditional batch-normalization with VAEs, we propose conditional weight normalization (CWN), see below for details. Apart from this adaptation, the architecture is the same as in Section 5.1.

The first metric, GAN-test, is obtained by training a classifier on natural image data and evaluating it on generated samples. It is sensitive to sample quality only. Our AV-ADE model obtains a slightly higher GAN-test score, suggesting comparable sample quality, which is in line with the results in Section 5.1.

model	GAN-test (%)	GAN-train (%)
GAN	71.8	29.7
AV-ADE	**76.9**	**73.4**
DCGAN†	58.2	65.0

Table 8: GAN-test and GAN-train measures for class conditional CQFG and GAN models on CIFAR-10. The performance of the DCGAN† model, though not directly comparable, is provided as a reference point.

The second metric, GAN-train, is obtained by training a classifier on generated samples and evaluating it on natural images. Having established similar GAN-test performance, this demonstrates improved sample diversity of the AV-ADE model and shows that the coverage-driven training improves the support of the learned model.

Class conditioning with conditional weight-normalization. To perform this evaluation we develop a class conditional version of our AV-ADE model. The discriminator is conditioned using the class conditioning introduced by Miyato and Koyama [36]. GAN generators are typically made class-conditional using conditional batch normalization [8, 13], however batch normalization is

known to be detrimental in VAEs [26], as we verified in practice. To address this issue, we propose conditional weight normalization (CWN). As in weight normalization [44], we separate the training of the scale and the direction of the weight matrix. Additionally, the scaling factor $g(y)$ of the weight matrix $\mathbf{v}$ is conditioned on the class label y:

$$\mathbf{w} = \frac{g(y)}{\|\mathbf{v}\|}\mathbf{v}, \tag{16}$$

We also make the network biases conditional on the class label. Otherwise, the architecture is the same one used for the experiments in Section 5.1.

E.2 Model evaluation using precision and recall

In this section, we evaluate our models using the precision and recall procedure of Sajjadi et al. [43]. This evaluation is relevant as it seeks to evaluate coverage of the support and the quality of the samples separately, rather than aggregating them into a single score. Intuitively, the precision measures the sample quality of the model, while the recall measures to what extent the model covers the support of the target distribution. The precision-recall metrics are determined based on a clustering of *Pool3* features extracted from training images and generated images using a pre-trained Inception network. The resulting histograms over cluster assignments are then used to assess the similarity of the distributions.

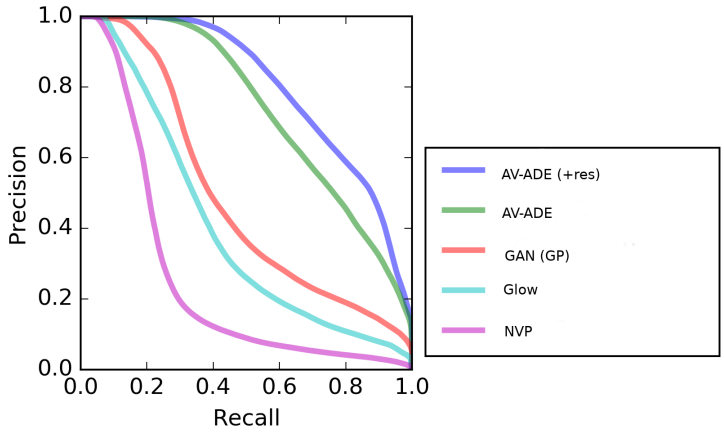


Figure 19: Precision-recall curves using the evaluation procedure of [43].

Figure 19 presents the evaluation of our models in Section 5.1, as well as the Glow and Real-NVP models, using the official code provided online by the authors at `https://github.com/msmsajjadi/precision-recall-distributions`. Our AV-ADE model obtains a better area under curve (AUC) than the GAN baseline, and model refinements improve AUC further. For comparison, Glow and Real-NVP have lower AUC than both GAN and our models.

F Qualitative influence of the feature space flexibility

We experiment with different architectures to implement the invertible mapping used to build the feature space as presented in Section 4. To assess the impact of the expressiveness of the invertible model on the behavior of our framework, we modify various standard parameters of the architecture. Popular invertible models such as Real-NVP [9] readily offer the possibility of extracting latent representation at several scales, separating global factors of variations from low level detail. Thus, we experiment with varying number of scales. Another way of increasing the flexibility of the model is to change the number of residual blocks used in each invertible layer. Note that all the models evaluated so far in the main body of the paper are based on a single scale and two residual blocks, except the one denoted with (s2). In addition to our AV-ADE models, we also compare with similar models trained with maximum likelihood estimation (V-ADE). Models are first trained with maximum-likelihood estimation, then with both coverage and quality driven criteria.

The results in Table 9 show that factoring out features at two scales rather than one is helpful in terms of BPD. For the AV-ADE models, however, IS and FID deteriorate with more scales, and so a tradeoff between must be struck. For the V-ADE models, the visual quality of samples also improves when using multiple scales, as reflected in better IS and FID scores. Their quality, however, remains far worse than those produced with the coverage and quality training used for the AV-ADE models. Samples in the maximum-likelihood setting are provided in Figure 20. With three or more scales, models exhibit symptoms of overfitting: train BPD keeps decreasing while test BPD starts increasing, and IS and FID also degrade.

Scales	Blocks	BPD ↓	IS ↑	FID ↓
1	2	3.77	**7.9**	**20.1**
2	2	3.48	6.9	27.7
2	4	**3.46**	6.9	28.9
3	3	3.49	6.5	31.7

(a) AV-ADE models

Scales	Blocks	BPD ↓	IS ↑	FID ↓
1	2	3.52	3.0	112.0
2	2	**3.41**	**4.5**	85.5
3	2	3.45	4.4	**78.7**
4	1	3.49	4.1	82.4

(b) V-ADE models

Table 9: Evaluation on CIFAR-10 of different architectures of the invertible layers of the model.

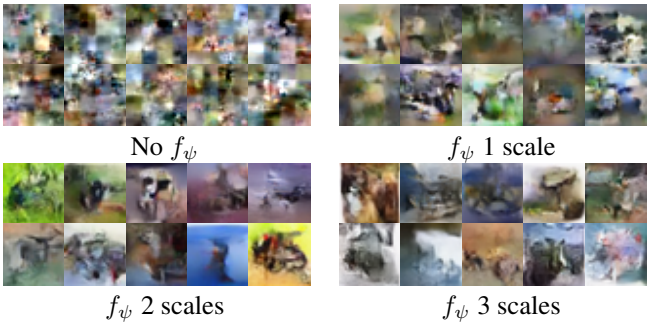


Figure 20: Samples obtained using VAE models trained with MLE (Table 9b) showing qualitative influence of multi-scale feature space. The models include one without invertible decoder layers, and with real-NVP layers using one, two and three scales. The samples illustrate the impact of using invertible real-NVP layers in these autoencoders.

G Visualisations of reconstructions

We display reconstructions obtained by encoding and then decoding ground truth images with our models (AV-ADE from Section 5.1) in Figure 21. As is typical for expressive variational autoencoders, real images and their reconstructions cannot be distinguished visually.

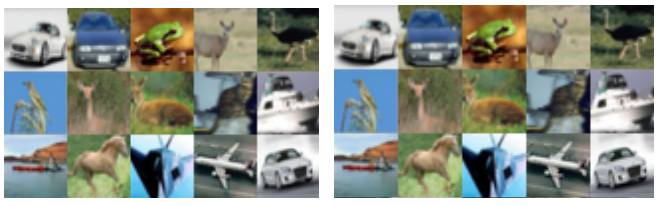


Figure 21: Real images and their reconstructions with the AV-ADE models.

H Coverage and Quality driven training algorithm

Algorithm 1 Coverage and Quality driven training for our AV-ADE model.

for number of training steps **do**

- Sample m real images $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from p^*, approximated by the dataset.
- Map the real images to feature space $\{f(\boldsymbol{x}^{(1)}), \ldots, f(\boldsymbol{x}^{(m)})\}$ using the invertible transformation f.
- Encode the feature space vectors using the VAE encoder and get parameters for the posterior $q_\phi(\boldsymbol{z}|f(\boldsymbol{x}))$.
- Sample m latent variable vectors, $\{\hat{\boldsymbol{z}}^{(1)}, \ldots, \hat{\boldsymbol{z}}^{(m)}\}$ from the posterior $q_\phi(z|x)$, and m latent variable vectors $\{\tilde{\boldsymbol{z}}^{(1)}, \ldots, \tilde{\boldsymbol{z}}^{(m)}\}$ from the VAE prior $p_\theta(z)$
- Decode both sets of latent variable vectors using the VAE decoder into the means of conditional Gaussian distributions, $\{\mu(\hat{\boldsymbol{z}})^{(1)}, \ldots, \mu(\hat{\boldsymbol{z}})^{(m)}\}$ and $\{\mu(\tilde{\boldsymbol{z}})^{(1)}, \ldots, \mu(\tilde{\boldsymbol{z}})^{(m)}\}$
- Sample from the Gaussian densities obtained, $\{\mathcal{N}(.|\mu(\hat{\boldsymbol{z}})^{(i)}, \sigma I_n)\}_{i\leq m}$ and $\{\mathcal{N}(.|\mu(\tilde{\boldsymbol{z}})^{(i)}, \sigma I_n)\}_{i\leq m}$, which yields reconstructions in feature space $\{\widehat{f(\boldsymbol{x})}^{(i)}\}_{i\leq m}$ and samples in feature space $\{\widetilde{f(\boldsymbol{x})}^{(i)}\}_{i\leq m}$
- Map the samples and reconstructions back to image space using the inverse of the invertible transformation f^{-1} which yields reconstructions $\{\hat{\boldsymbol{x}}^{(i)}\}_{i\leq m}$ and samples $\{\tilde{\boldsymbol{x}}^{(i)}\}_{i\leq m}$
- Compute $\mathcal{L}_{\mathrm{C}}(p_\theta)$ using ground truth images $\{\boldsymbol{x}^{(i)}\}_{i\leq m}$ and their reconstructions $\{\hat{\boldsymbol{x}}^{(i)}\}_{i\leq m}$
- Compute $\mathcal{L}_{\mathrm{Q}}(p_\theta)$ by feeding the ground truth images $\{\boldsymbol{x}^{(i)}\}_{i\leq m}$ together with the sampled images $\{\tilde{\boldsymbol{x}}^{(i)}\}_{i\leq m}$ to the discriminator
- Optimize the discriminator by gradient descent to bring $\mathcal{L}_Q$ closer to $\mathcal{L}_{\mathrm{Q}}^*$
- Optimize the generator by gradient descent to minimize $\mathcal{L}_{\mathrm{Q}} + \mathcal{L}_{\mathrm{C}}$

end for