GRADIENT ORIGIN NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper proposes a new type of generative model that is able to quickly learn a latent representation without an encoder. This is achieved by initialising a latent vector with zeros, then using the negative gradient of the data fitting loss with respect to this zero vector as new latent points. The approach has similar characteristics to autoencoders, but with a simpler architecture, and is demonstrated in a variational autoencoder equivalent that permits sampling. This also allows implicit representation networks to learn a space of implicit functions without requiring a hypernetwork, retaining their representation advantages across datasets. The experiments show that the proposed method converges faster, with significantly lower reconstruction error than autoencoders, while requiring half the parameters.

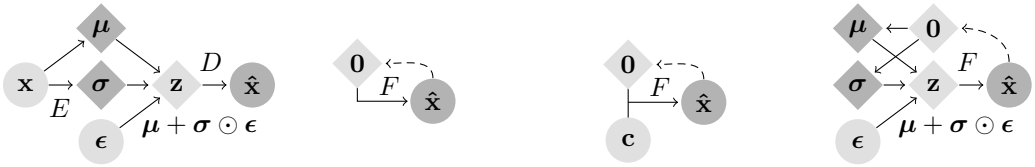
1 INTRODUCTION

Observable data in nature has some parameters which are known, such as local coordinates, but also some unknown parameters such as how the data is related to other examples. Generative models, which learn a distribution over observables, are central to our understanding of patterns in nature and allow for efficient query of new unseen examples. In recent years, deep generative models have received significant interest due to their ability to capture a broad set of features when modelling data distributions. As such, they offer direct applications such as synthesising high fidelity images (Karras et al., 2020), super-resolution (Dai et al., 2019), speech synthesis (Li et al., 2019), and drug discovery (Segler et al., 2018), as well as benefits for downstream tasks including semi-supervised learning (Chen et al., 2020) and anomaly detection (Gong et al., 2019).

A number of methods have been proposed such as Variational Autoencoders (VAEs, Figure 1a), which learn to encode the data to a latent space that follows a normal distribution permitting sampling (Kingma & Welling, 2014). Generative Adversarial Networks (GANs) have two competing networks, one which generates data and another which discriminates from implausible results (Goodfellow et al., 2014). Variational approaches that approximate the posterior using gradient descent (Lipton & Tripathi, 2017) and short run MCMC (Nijkamp et al., 2020) respectively have been proposed, but to obtain a latent vector for a sample, they require iterative gradient updates. Autoregressive Models (Van Den Oord et al., 2016) decompose the data distribution as the product of conditional distributions and Normalizing Flows (Rezende & Mohamed, 2015) chain together invertible functions; both methods allow exact likelihood inference. Energy-Based Models (EBMs) map data points to energy values proportional to likelihood thereby permitting sampling through the use of Monte Carlo Markov Chains (Du & Mordatch, 2019). In general to support encoding, these approaches require separate encoding networks, are limited to invertible functions, or require multiple sampling steps.

Implicit representation learning (Park et al., 2019; Tancik et al., 2020), where a network is trained on data parameterised continuously rather than in discrete grid form, has seen a surge of interest due to the small number of parameters, speed of convergence, and ability to model fine details. In particular, sinusoidal representation networks (SIRENs) (Sitzmann et al., 2020b) achieve impressive results, modelling complex signals with high precision, thanks to their use of periodic activations paired with carefully initialised MLPs. So far, however, these models have been limited to modelling single data samples, or use an additional hypernetwork or meta learning (Sitzmann et al., 2020a) to estimate the weights of a simple implicit model, which adds significant complexity.

This paper proposes Gradient Origin Networks (GONs), which are a new type of generative model (Figure 1b) that do not require encoders or additional hypernetworks. This is achieved by initialising latent points at the origin, then using the negative gradient of the data fitting loss with respect to these



(a) VAE (b) GON (c) Implicit GON (d) Variational GON

Figure 1: Gradient Origin Networks (GONs; b) use negative gradients (dashed lines) as encodings meaning that only a single network F is required, which can optionally be an implicit representation network (c). Unlike VAEs (a) which use two networks, E and D, variational GONs (d) permit sampling but do not require a secondary encoding network.

points as the latent space. At inference, the latent vector can be obtained in a single step without requiring iteration; GONs are shown to have similar characteristics to convolutional autoencoders and variational autoencoders while requiring approximately half the parameters. This is also shown to work well at learning a space of implicit functions using implicit representation networks (such as SIRENs), resulting in a simpler overall architecture.

2 METHOD

Consider some dataset $\mathbf{x} \sim p_d$ of continuous or discrete signals $\mathbf{x} \in \mathbb{R}^m$, it is typical to assume that the data can be represented by low dimensional latent variables $\mathbf{z} \in \mathbb{R}^k$, which can be used by a generative neural network to reconstruct the data. These variables are often estimated through the use of a secondary encoding network; however, for any arbitrary input, the gradients of the network contain substantial information about data points and can thus be used as encodings, making additional networks redundant. We formally define this process for a neural network $F\colon \mathbb{R}^k \to \mathbb{R}^m$ and reconstruction loss $\mathcal{L}(\mathbf{a}, \mathbf{b})$ that measures the distance between $\mathbf{a}$ and $\mathbf{b}$: latent vectors are obtained by first initialising at the origin, $\mathbf{z}_0 = \mathbf{0}$, then the negative gradient of the loss with respect to $\mathbf{z}_0$ is calculated $\mathbf{z} = -\nabla_{\mathbf{z}_0}\mathcal{L}(\mathbf{x}, F(\mathbf{z}_0))$. This process is jointly optimised with the reconstructed data, $\hat{\mathbf{x}} = F(\mathbf{z})$, by minimising the Gradient Origin Network loss function:

$$G_{\mathbf{x}} = \mathcal{L}(\mathbf{x}, F(-\nabla_{\mathbf{z}_0}\mathcal{L}(\mathbf{x}, F(\mathbf{z}_0)))). \tag{1}$$

The computation of $\mathbf{z}$ can intuitively be interpreted as finding a latent vector that minimises the reconstruction loss through gradient descent. A single step, however, is sufficient in part due to the joint optimisation procedure, the determinism associated with the constant initialisation of $\mathbf{z}_0$, and the freedom within the latent space dimensions. This gradient computation thereby plays a similar role to an encoder, while F can be viewed as a decoder, with the outer loss term determining the overall reconstruction quality. The use of a single network to perform both roles has the advantage of simplifying the overall architecture, avoiding the need to balance networks, and avoiding bottlenecks; this is demonstrated in Figure 1b which provides a visualisation of the GON process.

To permit sampling, the variational approach can be applied naturally to GONs. Specifically, the reparameterisation trick for a normally distributed prior as proposed by Kingma & Welling (2014) is applied to the latent space: $\mathbf{z}' = \mu(\mathbf{x}) + \sigma(\mathbf{x}) \odot \boldsymbol{\epsilon}$ where the mean and standard deviation, μ and σ respectively, are linear transformations of $\mathbf{z}$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. Consequently, by choosing $\mathcal{L}$ as the negative variational lower bound, the log-likelihood of the data can be maximised:

$$\mathcal{L}^{\text{VAE}} = D_{\text{KL}}(\mathcal{N}(\mu(\mathbf{x}), \sigma(\mathbf{x}))||\mathcal{N}(\mathbf{0}, \boldsymbol{I})) + \mathcal{L}^{\text{BCE}}(F(\mathbf{z}'), \mathbf{x}), \tag{2}$$

where D_{KL} is the Kullback–Leibler divergence and $\mathcal{L}^{\text{BCE}}$ is the binary cross entropy function. This is then substituted into Equation 1 replacing both the inner and outer $\mathcal{L}$ functions. These steps are shown in Figure 1d, which in practice has a simple implementation since it only requires a single network.

In implicit representation networks, data samples $\mathbf{x} = \{(\mathbf{c}, \Phi_{\mathbf{x}}(\mathbf{c}))\}$ are represented in terms of coordinates $\mathbf{c} \in \mathbb{R}^n$ and corresponding data $\Phi\colon \mathbb{R}^n \to \mathbb{R}^m$. The concatenation of latent variables with coordinates can therefore be seen as points on a manifold that describes the full dataset in keeping with the manifold hypothesis (Fefferman et al., 2016). An implicit Gradient Origin Network

can be trained on this space to mimic Φ using a neural network $F\colon \mathbb{R}^{n+k} \rightarrow \mathbb{R}^m$, thereby learning a space of implicit functions by modifying Equation 1:

$$I_{\mathbf{x}} = \int \mathcal{L}\Big(\Phi_{\mathbf{x}}(\mathbf{c}), F\Big(\mathbf{c} \oplus -\nabla_{\mathbf{z}_0} \int \mathcal{L}\big(\Phi_{\mathbf{x}}(\mathbf{c}), F(\mathbf{c} \oplus \mathbf{z}_0)\big) d\mathbf{c}\Big)\Big) d\mathbf{c}, \tag{3}$$

where both integrations are performed over the space of coordinates and $\mathbf{c} \oplus \mathbf{z}$ represents a concatenation (Figure 1c). Similar to the non-implicit approach, the computation of latent vectors can be expressed as $\mathbf{z} = -\nabla_{\mathbf{z}_0} \int \mathcal{L}\big(\Phi_{\mathbf{x}}(\mathbf{c}), F(\mathbf{c} \oplus \mathbf{z}_0)\big) d\mathbf{c}$. In particular, we parameterise F with a SIREN (Sitzmann et al., 2020b), finding that it is capable of modelling both high and low frequency components in high dimensional spaces.

There are a number of interesting generalisations that make this approach applicable to other tasks. In Equations 1 and 3 we use the same $\mathcal{L}$ in both the inner term and outer term, however it is possible to use different functions; through this, training a GON concurrently as a generative model and classifier is possible, or through some minor modifications to the loss, solely as a classifier:

$$C_{\mathbf{x}} = \mathcal{L}^{\text{CCE}}(f(-\nabla_{\mathbf{z}_0} \mathcal{L}(\mathbf{x}, F(\mathbf{z}_0))), \mathbf{y}), \tag{4}$$

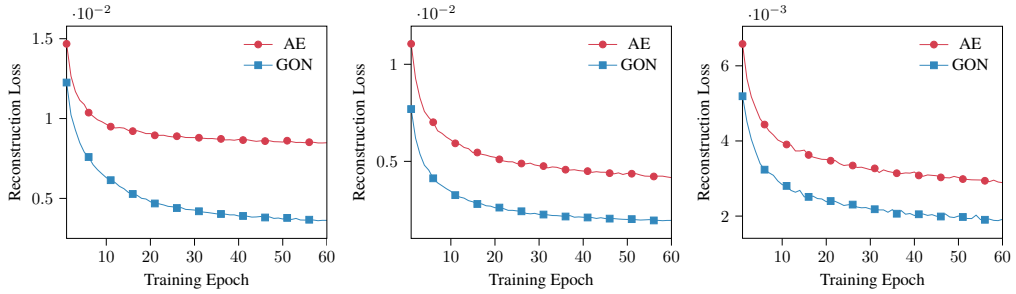
where $\mathbf{y}$ are the target labels, f is a single linear layer followed by softmax, and $\mathcal{L}^{\text{CCE}}$ is the categorical cross entropy loss function. Another possibility is modality conversion for translation tasks; in this case the inner reconstruction loss is performed on the source signal and the outer loss on the target signal.

3 Results & Discussion

We evaluate Gradient Origin Networks on a variety of image datasets: MNIST, Fashion-MNIST (Xiao et al., 2017), NORB (LeCun et al., 2004), COIL-20 (Nane et al., 1996), and CIFAR-10 (Krizhevsky et al., 2009). Simple models are used: implicit GONs consist of approximately 4 hidden layers of 256 units and convolutional GONs consist of 4 convolution layers with Batch Normalization (Ioffe & Szegedy, 2015) and the ELU non-linearity (Clevert et al., 2016); both are trained with the Adam optimiser (Kingma & Ba, 2015). All graphs shown in this paper are the average of 3 runs.

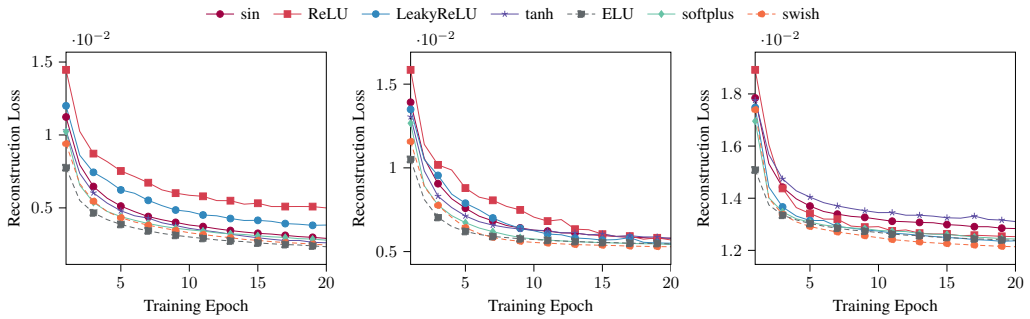
An ablation study is performed, comparing convolutional GONs with autoencoders whose decoders have exactly the same architecture as the GONs (Figure 2) and where the autoencoder decoders mirror their respective encoders. The mean reconstruction loss over the training set is measured after each epoch for a variety of latent space sizes. Despite having fewer than half the number of parameters, we find that GONs achieve significantly lower reconstruction loss over a wide range of architectures. Further experiments evaluating the impact of latent space size and model capacity are performed in Appendix A.

Figure 3 demonstrates the effect activation functions have on convolutional GON performance for different numbers of latent variables. Since optimising GONs requires computing second order derivatives, the choice of nonlinearity requires different characteristics to standard models. In particular, GONs prefer functions that are not only effective activation functions, but also whose first and second derivatives are nonlinear functions, unlike ReLUs where $\text{ReLU}''(x) = 0$. The ELU non-linearity is effective with all tested architectures.



(a) 1x1x512 latent variables (b) 2x2x512 latent variables (c) 4x4x512 latent variables

Figure 2: Gradient Origin Networks trained on CIFAR-10 are found to outperform autoencoders using exactly the same architecture without the encoder, requiring half the number of parameters.

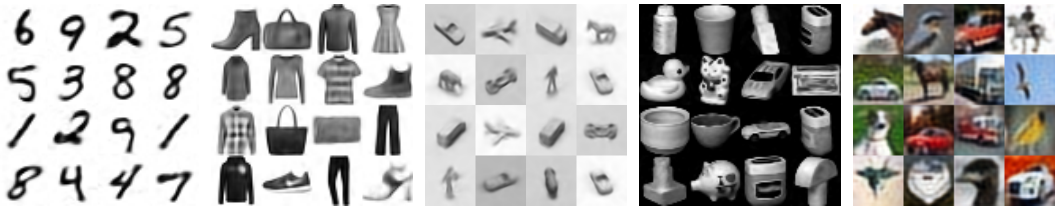


(a) 512 latent variables (b) 128 latent variables (c) 32 latent variables

Figure 3: The impact of activation function and number of latent variables on model performance for a GON measured by comparing reconstruction losses through training.

The representation ability of implicit GONs is shown in Figure 4 where we overfit to large image datasets using a relatively small number of parameters. In particular, Figure 4a shows MNIST can be well-represented with just 4,385 parameters (a SIREN with 3 hidden layers each with 32 hidden units, and 32 latent dimensions). An advantage of modelling data with implicit networks is that coordinates can be at arbitrarily high resolutions. In Figure 5 we train on 32x32 images then reconstruct at 256x256. A significant amount of high frequency detail is modelled despite only seeing low resolution images. The structure of the implicit GON latent space is shown by sampling latent codes from pairs of images, and then spherically interpolating between them to synthesise new samples (Figure 6). These samples are shown to capture variations in shape (the shoes in Figure 6b), size, and rotation (Figure 6d).

We assess the quality of samples from variational GONs using convolutional models trained to convergence in Figure 7. These are diverse and often contain fine details. Samples from an implicit GON trained with early stopping, as a simple alternative, can be found in Appendix C however this approach results in fine details being lost. GONs are also found to converge quickly; we plot reconstructions at multiple time points during the first minute of training (Figure 8). After only 3 seconds of training on a single GPU, a substantial amount of signal information from MNIST is modelled.



(a) 4,385 parameters (b) 297k parameters (c) 270k parameters (d) 270k parameters (e) 396k parameters

Figure 4: Overfitting implicit GONs with few parameters demonstrates their representation ability.

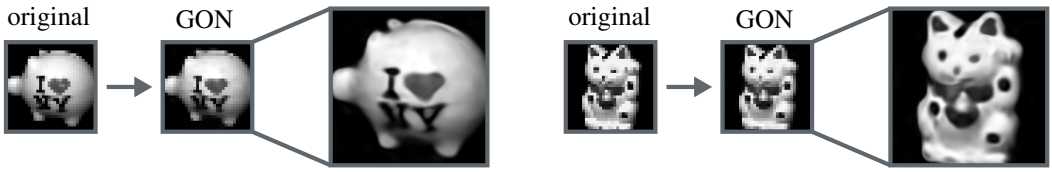
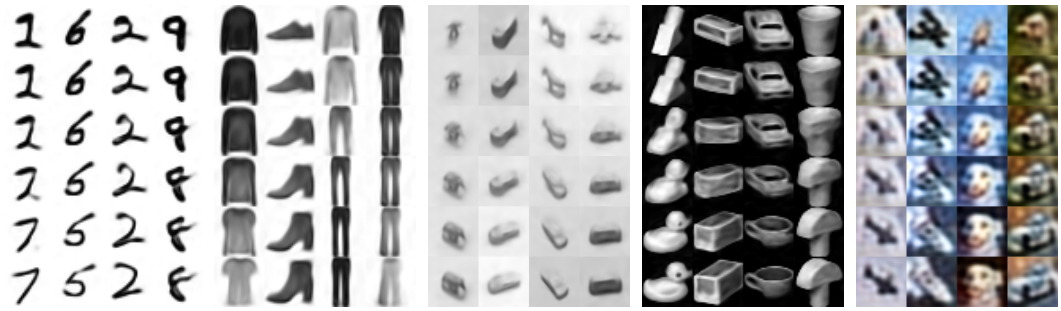


Figure 5: By training an implicit GON on 32x32 images, then sampling at 256x256, super-resolution is possible despite never observing high resolution data.

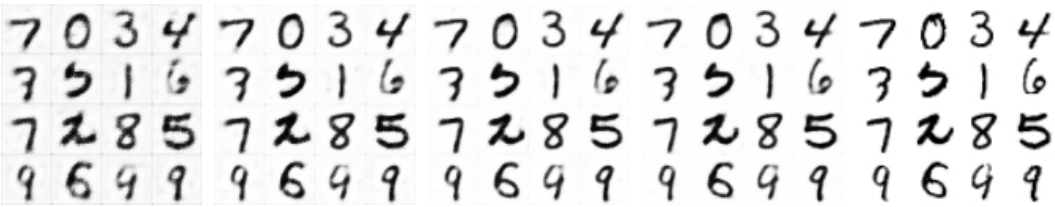


(a) MNIST (b) Fashion-MNIST (c) Small NORB (d) COIL20 (e) CIFAR-10

Figure 6: Spherical linear interpolations between points in the latent space for trained implicit GONs using different datasets (approximately 2-10 minutes training per dataset on a single GPU).

(a) 4,331 parameters (b) 270k parameters (c) 270k parameters (d) 270k parameters (e) 270k parameters

Figure 7: Random samples drawn from a normally distributed latent space for a variational GON.



(a) 3s 300 steps (b) 10s 1000 steps (c) 30s 3000 steps (d) 60s 6000 steps (e) Ground Truth

Figure 8: Convergence of convolutional GONs with 74k parameters.

Despite similarities with autoencoder approaches, the absence of an encoding network offers several advantages. VAEs with overpowered decoders are known to ignore the latent variables (Chen et al., 2017) whereas GONs only have one network that equally serves both encoding and decoding functionality. Designing inference networks for complex decoders is not a trivial task (Vahdat & Kautz, 2020), however, inferring latent variables using a GON simplifies this procedure. Similar to GONs, Normalizing Flow methods are also capable of encoding and decoding with a single set of weights, however, they achieve this by restricting the network to be invertible. This requires considerable architectural restrictions that affect performance, make them less parameter efficient, and are unable to reduce the dimension of the data (Kingma & Dhariwal, 2018). GONs on the other hand use negative gradients as encodings which allow for arbitrary functions to be used.

A number of previous works have used gradient-based computations to learn latent vectors however as far as we are aware, we are the first to use a single gradient step which is jointly optimised with the feedforward pass, making it fundamentally different to these approaches. Latent encodings have been estimated for pre-trained generative models without encoders, namely Generative Adversarial Networks, using approaches such as standard gradient descent (Lipton & Tripathi, 2017; Zhu et al., 2016). A number of approaches have trained generative models directly with gradient-based inference (Han et al., 2017; Bojanowski et al., 2018; Zadeh et al., 2019); these assign latent vectors to data

points and jointly learn them with the network parameters through standard gradient descent or Langevin dynamics. This is very slow, however, and convergence for unseen data samples is not guaranteed. Short run MCMC has also been applied (Nijkamp et al., 2020) however this still requires approximately 25 update steps. Since GONs train end-to-end, the optimiser can make use of the second order derivatives to allow for inference in a single step.

In the case of implicit GONs, the integration terms in Equation 3 result in computation time that scales in proportion with the data dimension. This makes training slower for very high dimensional data, although we have not yet investigated Monte Carlo integration. While learning rates are quite resilient, some tuning is required. However, in general GONs are stable, capable of generating quality samples with an exceptionally small number of parameters, and converge to diverse results with few iterations. Nevertheless, there are avenues to explore so as to improve the quality of samples and scale to larger datasets. In particular, it would be beneficial to focus on how to better sample these models, perform formal analysis on the gradients, and investigate whether the distance function could be improved to better capture fine details.

CONCLUSION

In conclusion, we have introduced a method which computes the negative gradient of the data fitting loss with respect to the origin, and then jointly fits the data while learning this new point of reference in the latent space. The results show that this approach is able to represent datasets using a small number of parameters with a simple overall architecture, which has advantages in applications such as implicit representation networks. GONs are shown to converge faster with lower overall reconstruction loss than autoencoders, using the exact same architecture but without the encoder. Experiments show that the choice of non-linearity is important, as the network derivative jointly acts as the encoding function.

REFERENCES

Piotr Bojanowski, Armand Joulin, David Lopez-Pas, and Arthur Szlam. Optimizing the Latent Space of Generative Networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, volume 80, pp. 600–609, 2018.

Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative Pretraining from Pixels. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational Lossy Autoencoder. In *5th International Conference on Learning Representations, ICLR*, 2017.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). In *4th International Conference on Learning Representations, ICLR*, 2016.

Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order Attention Network for Single Image Super-Resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11065–11074, 2019.

Yilun Du and Igor Mordatch. Implicit Generation and Generalization in Energy-Based Models. In *Neural Information Processing Systems (NeurIPS)*, 2019.

Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the Manifold Hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing Normality to Detect Anomaly: Memory-augmented Deep Autoencoder for Unsupervised Anomaly Detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1705–1714, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. 2014.

Tian Han, Yang Lu, Song Chun Zhu, and Ying Nian Wu. Alternating Back-Propagation for Generator Network. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 1976–1984, 2017.

Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*, pp. 448–456, 2015.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR*, 2015.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *2nd International Conference on Learning Representations, ICLR*, 2014.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative Flow with Invertible 1x1 Convolutions. In *Advances in neural information processing systems*, pp. 10215–10224, 2018.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning Multiple Layers of Features from Tiny Images. *Technical Report*, 2009.

Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, volume 2, pp. II–104, 2004.

Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. Neural Speech Synthesis with Transformer Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 6706–6713, 2019.

Zachary C Lipton and Subarna Tripathi. Precise Recovery of Latent Vectors from Generative Adversarial Networks. In *5th International Conference on Learning Representations, ICLR*, 2017.

SA Nane, SK Nayar, and H Murase. Columbia Object Image Library: COIL-20. *Technical Report CUCS-005-96, Columbia University*, 1996.

Erik Nijkamp, Bo Pang, Tian Han, Linqi Zhou, Song-Chun Zhu, and Ying Nian Wu. Learning Multi-layer Latent Variable Model via Variational Optimization of Short Run MCMC for Approximate Inference. *stat*, 1050:17, 2020.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 165–174, 2019.

Danilo Jimenez Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. *32nd International Conference on Machine Learning, ICML*, 2:1530–1538, 2015.

Marwin HS Segler, Mike Preuss, and Mark P Waller. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature*, 555(7698):604–610, 2018.

Vincent Sitzmann, Eric R. Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. MetaSDF: Meta-learning Signed Distance Functions. *arXiv preprint arXiv:2006.09662*, 2020a.

Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit Neural Representations with Periodic Activation Functions. *arXiv preprint arXiv:2006.09661*, 2020b.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Arash Vahdat and Jan Kautz. NVAE: A Deep Hierarchical Variational Autoencoder. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel Recurrent Neural Networks. In *33rd International Conference on Machine Learning, ICML 2016*, volume 4, pp. 2611–2620, 2016.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Amir Zadeh, Yao-Chong Lim, Paul Pu Liang, and Louis-Philippe Morency. Variational Auto-Decoder. *arXiv preprint arXiv:1903.00840*, 2019.

Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative Visual Manipulation on the Natural Image Manifold. In *European Conference on Computer Vision*, pp. 597–613, 2016.

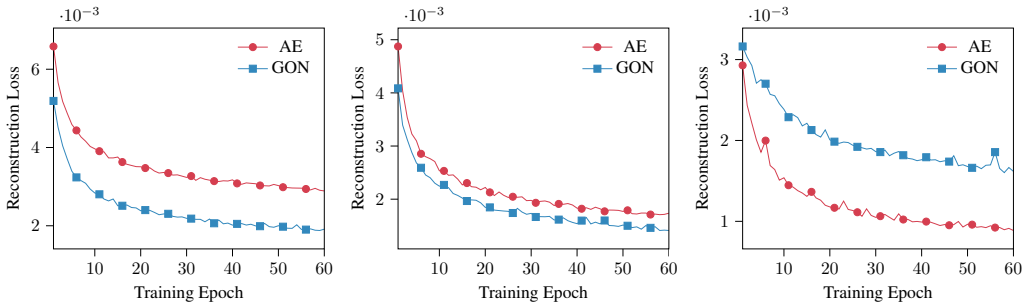
A Comparison with Autoencoders

This section compares Gradient Origin Networks with autoencoders that use exactly the same architecture, as described in the paper. All experiments are performed with the CIFAR-10 dataset. After each epoch the model is fixed and the mean reconstruction loss is measured over the training set.

In Figure 9 the depth of networks is altered by removing blocks thereby downscaling to various resolutions. We find that GONs outperform autoencoders until latents have a spatial size of 8x8 (where the equivalent GON now only has only 2 convolution layers). Considering the limit where neither model has any parameters, the latent space is the input data i.e. $\mathbf{z} = \mathbf{x}$. Substituting the definition of a GON (Equation 1) this becomes $-\nabla_{\mathbf{z_0}}\mathcal{L} = \mathbf{x}$ which simplifies to $\mathbf{0} = \mathbf{x}$ which is a contradiction. This is not a concern in normal practice, as evidenced by the results presented here.

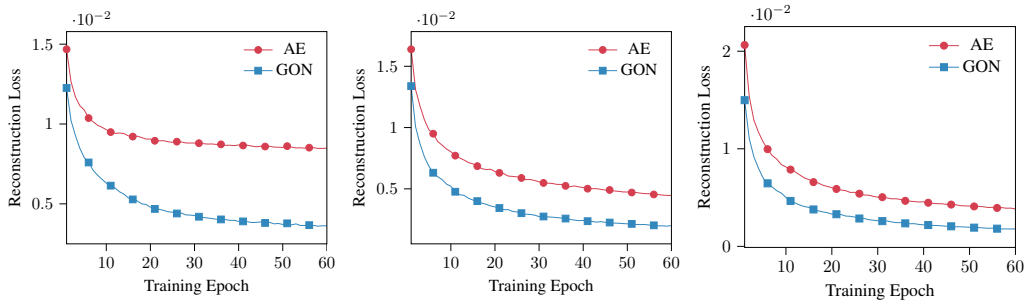
Figure 10 explores the relationship between autoencoders and GONs when changing the number of convolution filters; GONs are found to outperform autoencoders in all cases. The discrepancy between loss curves decreases as the number of filters increases likely due to the diminishing returns of providing more capacity to the GON when its loss is significantly closer to 0.

A similar pattern is found when decreasing the latent space (Figure 11); in this case the latent space likely becomes the limiting factor. With larger latent spaces GONs significantly outperform autoencoders, however, when the latent bottleneck becomes more pronounced this lead lessens.



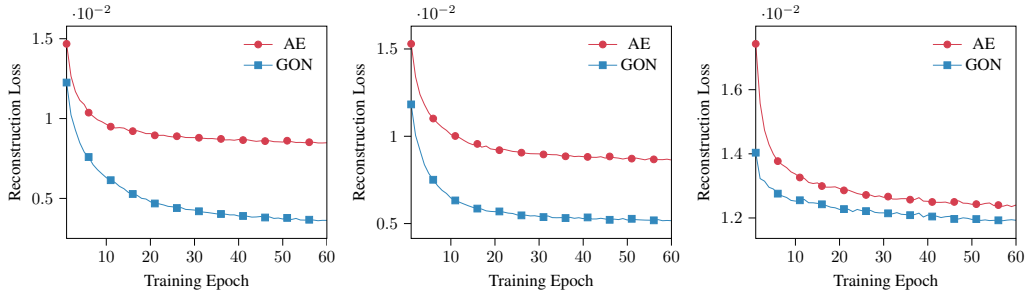
(a) 4x4x512 latent variables (b) 6x6x512 latent variables (c) 8x8x512 latent variables

Figure 9: Higher bandwidth latent spaces in convolutional GONs trained on CIFAR-10; at the limit autoencoders tend towards the identity function whereas GONs are unable to operate with no parameters. The GON uses exactly same architecture as the AE, without the encoder.



(a) $n_f = 16$ (b) $n_f = 64$ (c) $n_f = 128$

Figure 10: Capacities of GONs and autoencoders are increased by changing the number of convolution filters (each decoder/GON layer reduces the filters from $4n_f$ ending at n_f; the encoder mirrors this).

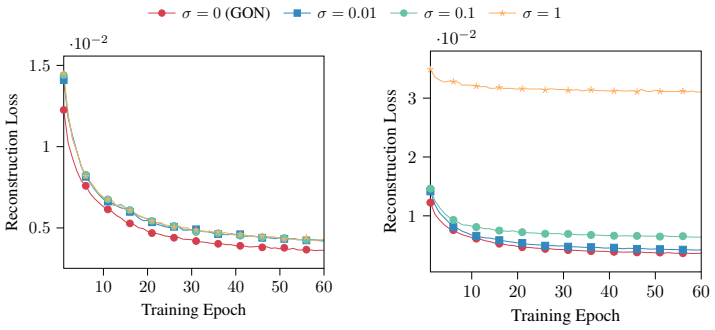


(a) 512 latent variables (b) 128 latent variables (c) 32 latent variables

Figure 11: Training loss plots comparing GONs and autoencoders for a variety of latent space sizes.

B INITIALISING $\mathbf{z}_0$ AT THE ORIGIN

We evaluate different initialisations of $\mathbf{z}_0$ in Figure 12 by sampling $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ for a variety of standard deviations σ. The proposed approach ($\sigma = 0$) achieves the lowest reconstruction loss (Figure 12a); results for $\sigma > 0$ are similar, suggesting that the latent space is adjusted so $\mathbf{z}_0$ simulates the origin. An alternative parameterisation of $\mathbf{z}$ is to use a use a single gradient descent style step $\mathbf{z} = \mathbf{z}_0 - \nabla_{\mathbf{z}_0} \mathcal{L}$ (Figure 12b), however, losses are higher than the proposed GON initialisation.



(a) $\mathbf{z} = -\nabla_{\mathbf{z}_0} \mathcal{L}$ (b) $\mathbf{z} = \mathbf{z}_0 - \nabla_{\mathbf{z}_0} \mathcal{L}$

Figure 12: Training GONs with $\mathbf{z}_0$ sampled from a variety of normal distributions with different standard deviations σ, $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$. Approach (a) directly uses the negative gradients as encodings while approach (b) performs one gradient descent style step initialised at $\mathbf{z}_0$.

C LATENT SPACE AND EARLY STOPPING

Figure 13 shows a GON trained as a classifier where the latent space is squeezed into 2D for visualisation (Equation 4) and Figure 14 shows that the negative gradients of a GON after 800 steps of training are approximately normally distributed; Figure 15 shows that the latents of a typical convolutional VAE after 800 steps are significantly less normally distributed than GONs.

To obtain new samples with implicit GONs, we can use early stopping as a simple alternative to variational approaches. These samples (Figure 16) are diverse and capture the object shapes.

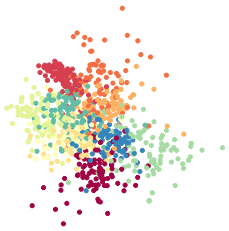


Figure 13: 2D latent space samples of an implicit GON classifier trained on MNIST (class colours).

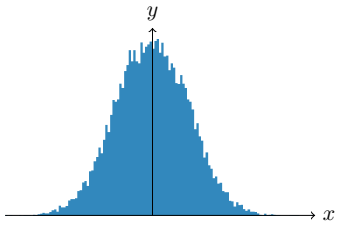


Figure 14: Histogram of latent gradients after 800 implicit GON steps with a SIREN.

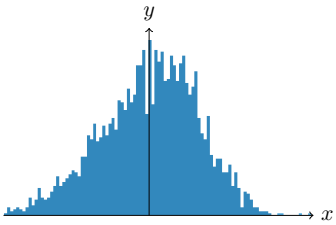


Figure 15: Histogram of traditional VAE latents after 800 steps.

(a) 4,385 parameters (b) 70k parameters (c) 270k parameters (d) 270k parameters (e) 270k parameters

Figure 16: GONs trained with early stopping can be sampled by approximating their latent space with a multivariate normal distribution. These images show samples from an implicit GON trained with early stopping.

D SUPER-RESOLUTION

Additional super-sampled images obtained by training an implicit GON on 28x28 MNIST data then reconstructing at a resolution of 256x256 are shown in Figure 17.

(a) Original data 28x28 (b) Implicit GON at 28x28

Figure 17: Super-sampling 28x28 MNIST at 256x256 coordinates using an implicit GON.