

Open Questions about Generative Adversarial Networks

What we'd like to find out about GANs that we don't know yet.

- Problem 1 What are the trade-offs between GANs and other generative models?
- Problem 2 What sorts of distributions can GANs model?
- Problem 3 How can we Scale GANs beyond image synthesis?
- Problem 4 What can we say about the global convergence of the training dynamics?
- Problem 5 How should we evaluate GANs and when should we use them?
- Problem 6 How does GAN training scale with batch size?
- Problem 7 What is the relationship between GANs and adversarial examples?

AUTHORS

Augustus Odena

PUBLISHED

April 9, 2019

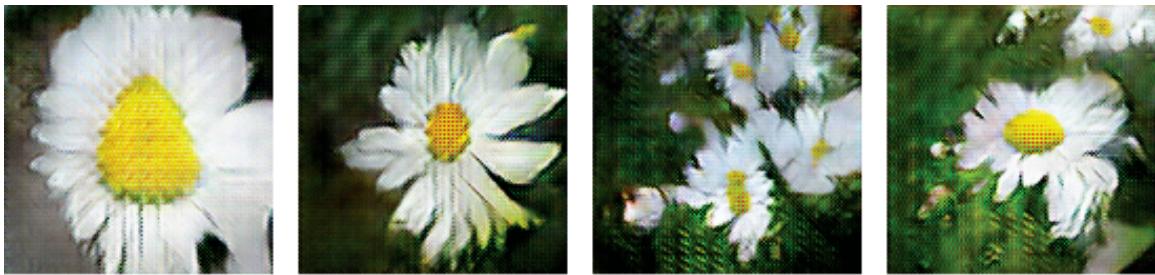
AFFILIATIONS

Google Brain Team

DOI

10.23915/distill.00018

By some metrics, research on Generative Adversarial Networks (GANs) has progressed substantially in the past 2 years. Practical improvements to image synthesis models are being made [1, 2, 3, 4, 5] almost too quickly to keep up with:



Odena et al.,
2016 [1]



Miyato et al.,
2017 [3]



Zhang et al.,
2018 [2]



Brock et al.,
2018 [4]

However, by other metrics, less has happened. For instance, there is still widespread disagreement about how GANs should be evaluated. Given that current image synthesis benchmarks seem somewhat saturated, we think now is a good time to reflect on research goals for this sub-field.

Lists of open problems have helped other fields with this [6, 7, 8, 9]. This article suggests open research problems that we'd be excited for other researchers to work on.¹ We assume a fair amount of background (or willingness to look things up) because we reference too many results to explain all those results in detail.

What are the Trade-Offs Between GANs and other Generative Models?

In addition to GANs, two other types of generative model are currently popular: Flow Models and

Autoregressive Models². Roughly speaking, Flow Models [11, 12, 13, 14] apply a stack of invertible transformations to a sample from a prior so that exact log-likelihoods of observations can be computed. On the other hand, Autoregressive Models [15, 16, 17, 18] factorize the distribution over observations into conditional distributions and process one component of the observation at a time (for images, they may process one pixel at a time.) Recent research [13, 19] suggests that these models have different performance characteristics and trade-offs. We think that accurately characterizing these trade-offs and deciding whether they are intrinsic to the model families is an interesting open question.

For concreteness, let's temporarily focus on the difference in computational cost between GANs and Flow Models. At first glance, Flow Models seem like they might make GANs unnecessary. Flow Models allow for exact log-likelihood computation and exact inference, so if training Flow Models and GANs had the same computational cost, GANs might not be useful. A lot of effort is spent on training GANs, so it seems like we should care about whether Flow Models make GANs obsolete³.

However, there seems to be a substantial gap between the computational cost of training GANs and Flow Models. To estimate the magnitude of this gap, we can consider two models trained on datasets of human faces. The GLOW model [13] is trained to generate 256x256 celebrity faces using 40 GPUs for 2 weeks and about 200 million parameters. In contrast, progressive GANs [19] are trained on a similar face dataset with 8 GPUs for 4 days, using about 46 million parameters, to generate 1024x1024 images. Roughly speaking, the Flow Model took 17 times more GPU days and 4 times more parameters to generate images with 16 times fewer pixels. This comparison isn't perfect,⁴ but it gives you a sense of things.

Why are the Flow Models less efficient? We see two possible reasons: First, maximum likelihood training might be computationally harder to do than adversarial training. In particular, if any element of your training set is assigned zero probability by your generative model, you will be penalized infinitely harshly! A GAN generator, on the other hand, is only penalized indirectly for assigning zero probability to training set elements, and this penalty is less harsh. Second, normalizing flows might be an inefficient way to represent certain functions. Section 6.1 of [20] does some small experiments on expressivity, but at present we're not aware of any in-depth analysis of this question.

We've discussed the trade-off between GANs and Flow Models, but what about Autoregressive Models? It turns out that Autoregressive Models can be expressed as Flow Models (because they are both reversible [14]) that are not parallelizable.⁵ It also turns out that Autoregressive Models are more time and parameter efficient than Flow Models. Thus, GANs are parallel and efficient but not reversible, Flow Models are reversible and parallel but not efficient, and Autoregressive models are reversible and efficient, but not parallel.

	Parallel	Efficient	Reversible
GANs	Yes	Yes	No
Flow Models	Yes	No	Yes
Autoregressive Models	No	Yes	Yes

This brings us to our first open problem:

Problem 1 What are the fundamental trade-offs between GANs and other generative models?

In particular, can we make some sort of CAP Theorem [22] type statement about reversibility, parallelism, and parameter/time efficiency?

One way to approach this problem could be to study more models that are a hybrid of multiple model families. This has been considered for hybrid GAN/Flow Models [23, 24], but we think that this approach is still underexplored.

We're also not sure about whether maximum likelihood training is necessarily harder than GAN training. It's true that placing zero mass on a training data point is not explicitly prohibited under the GAN training loss, but it's also true that a sufficiently powerful discriminator will be able to do better than chance if the generator does this. It does seem like GANs are learning distributions of low support in practice [25] though.

Ultimately, we suspect that Flow Models are fundamentally less expressive per-parameter than arbitrary decoder functions, and we suspect that this is provable under certain assumptions.

What Sorts of Distributions Can GANs Model?

Most GAN research focuses on image synthesis. In particular, people train GANs on a handful of standard (in the Deep Learning community) image datasets: MNIST [26], CIFAR-10 [27], STL-10 [28], CelebA [29], and Imagenet [30]. There is some folklore about which of these datasets is 'easiest' to model. In particular, MNIST and CelebA are considered easier than Imagenet, CIFAR-10, or STL-10 due to being 'extremely regular' [31]. Others have noted that 'a high number of classes is what makes ImageNet synthesis difficult for GANs' [1]. These observations are supported by the empirical fact that the state-of-the-art image synthesis model on CelebA [5] generates images that seem substantially more convincing than the state-of-the-art image synthesis model on Imagenet [4].

However, we've had to come to these conclusions through the laborious and noisy process of trying to train GANs on ever larger and more complicated datasets. In particular, we've mostly studied how GANs perform on the datasets that happened to be laying around for object recognition.

As with any science, we would like to have a simple theory that explains our experimental observations. Ideally, we could look at a dataset, perform some computations without ever actually training a generative model, and then say something like 'this dataset will be easy for a GAN to model, but not a VAE'. There has been some progress on this topic [32, 33], but we feel that more can be done. We can now state the problem:

Problem 2 Given a distribution, what can we say about how hard it will be for a GAN to model that distribution?

We might ask the following related questions as well: What do we mean by 'model the distribution'? Are we satisfied with a low-support representation, or do we want a true density model? Are there distributions that a GAN can never learn to model? Are there distributions that are learnable for a GAN in principle, but are not efficiently learnable, for some reasonable model of resource-consumption? Are the answers to these questions actually any different for GANs than they are for other generative models?

We propose two strategies for answering these questions:

- **Synthetic Datasets** - We can study synthetic datasets to probe what traits affect learnability. For example, in [32] the authors create a dataset of synthetic triangles. We feel that this angle is under-explored. Synthetic datasets can even be parameterized by quantities of interest, such as connectedness or smoothness, allowing for systematic study. Such a dataset could also be useful for studying other types of generative models.
- **Modify Existing Theoretical Results** - We can take existing theoretical results and try to modify the assumptions to account for different properties of the dataset. For instance, we could take results about GANs that apply given unimodal data distributions and see what happens to them when the data distribution becomes multi-modal.

How Can we Scale GANs Beyond Image Synthesis?

Aside from applications like image-to-image translation [34] and domain-adaptation [35] most GAN successes have been in image synthesis. Attempts to use GANs beyond images have focused on three domains:

- **Text** - The discrete nature of text makes it difficult to apply GANs. This is because GANs rely on backpropagating a signal from the discriminator through the generated content into the generator. There are two approaches to addressing this difficulty. The first is to have the GAN act only on continuous representations of the discrete data, as in [36]. The second is use an actual discrete model and attempt to train the GAN using gradient estimation as in [37]. Other, more sophisticated treatments exist [38], but as far as we can tell, none of them produce results that are competitive (in terms of perplexity) with likelihood-based [39] language models.
- **Structured Data** - What about other non-euclidean structured data, like graphs? The study of this type of data is called geometric deep learning [40]. GANs have had limited success here, but so have other deep learning techniques, so it's hard to tell how much the GAN aspect matters. We're aware of one attempt to use GANs in this space [41], which has the generator produce (and the discriminator 'critique') random walks that are meant to resemble those sampled from a source graph.
- **Audio** - Audio is the domain in which GANs are closest to achieving the success they've enjoyed with images. The first serious attempt at applying GANs to unsupervised audio synthesis is [42], in which the authors make a variety of special allowances for the fact that they are operating on audio. More recent work suggests GANs can even outperform autoregressive models on some perceptual metrics [43].

Despite these attempts, images are clearly the easiest domain for GANs. This leads us to the statement of the problem:

Problem 3 How can GANs be made to perform well on non-image data?

Does scaling GANs to other domains require new training techniques, or does it simply require better implicit priors for each domain?

We expect GANs to eventually achieve image-synthesis-level success on other continuous data, but that it will require better implicit priors. Finding these priors will require thinking hard about what makes sense and is computationally feasible in a given domain.

For structured data or data that is not continuous, we're less sure. One approach might be to make both the generator and discriminator be agents trained with reinforcement learning. Making this approach work could require large-scale computational resources [44]. Finally, this problem may just require fundamental research progress.

What can we Say About the Global Convergence of GAN Training?

Training GANs is different from training other neural networks because we simultaneously optimize the generator and discriminator for opposing objectives. Under certain assumptions ⁶, this simultaneous optimization is locally asymptotically stable [45, 46].

Unfortunately, it's hard to prove interesting things about the fully general case. This is because the discriminator/generator's loss is a non-convex function of its parameters. But all neural networks have this problem! We'd like some way to focus on just the problems created by simultaneous optimization. This brings us to our question:

Problem 4 When can we prove that GANs are globally convergent?

Which neural network convergence results can be applied to GANs?

There has been nontrivial progress on this question. Broadly speaking, there are 3 existing techniques, all of which have generated promising results but none of which have been studied to completion:

- **Simplifying Assumptions** - The first strategy is to make simplifying assumptions about the generator and discriminator. For example, the simplified LGQ GAN [47] — linear generator, Gaussian data, and quadratic discriminator — can be shown to be globally convergent, if optimized with a special technique [48, 49] and some additional assumptions.⁷ As another example, [50, 51] show under different simplifying assumptions that certain types of GANs perform a mixture of moment matching and maximum likelihood estimation. It seems promising to gradually relax those assumptions to see what

happens. For example, we could move away from unimodal distributions [52]. This is a natural relaxation to study because ‘mode collapse’ is a standard GAN pathology.

- **Use Techniques from Normal Neural Networks** - The second strategy is to apply techniques for analyzing normal neural networks (which are also non-convex) to answer questions about convergence of GANs. For instance, it’s argued in [53] that the non-convexity of deep neural networks isn’t a problem,⁸ because low-quality local minima of the loss function become exponentially rare as the network gets larger. Can this analysis be ‘lifted into GAN space’? In fact, it seems like a generally useful heuristic to take analyses of deep neural networks used as classifiers and see if they apply to GANs.
- **Game Theory** - The final strategy is to model GAN training using notions from game theory [54, 55, 56]. These techniques yield training procedures that provably converge to some kind of approximate Nash equilibrium, but do so using unreasonably large resource constraints. The ‘obvious’ next step in this case is to try and reduce those resource constraints.

How Should we Evaluate GANs and When Should we Use Them?

When it comes to evaluating GANs, there are many proposals but little consensus. Suggestions include:

- **Inception Score and FID** - Both these scores [57, 58] use a pre-trained image classifier and both have known issues [59, 60]. A common criticism is that these scores measure ‘sample quality’ and don’t really capture ‘sample diversity’.
- **MS-SSIM** - [1] propose using MS-SSIM [61] to separately evaluate diversity, but this technique has some issues and hasn’t really caught on.
- **AIS** - [62] propose putting a Gaussian observation model on the outputs of a GAN and using annealed importance sampling [63] to estimate the log likelihood under this model, but [23] show that estimates computed this way are inaccurate in the case where the GAN generator is also a flow model⁹.
- **Geometry Score** - [64] suggest computing geometric properties of the generated data manifold and comparing those properties to the real data.
- **Precision and Recall** - [65] attempt to measure both the ‘precision’ and ‘recall’ of GANs.
- **Skill Rating** - [66, 67] have shown that trained GAN discriminators can contain useful information with which evaluation can be performed.

Those are just a small fraction of the proposed GAN evaluation schemes. Although the Inception Score and FID are relatively popular, GAN evaluation is clearly not a settled issue. Ultimately, we think that confusion about *how to evaluate GANs* stems from confusion about *when to use GANs*. Thus, we have bundled those two questions into one:

What should we use GANs for? If you want an actual density model, GANs probably aren't the best choice. There is now good experimental evidence that GANs learn a 'low support' representation of the target dataset [23, 24, 25], which means there may be substantial parts of the test set to which a GAN (implicitly) assigns zero likelihood.

Rather than worrying too much about this,¹⁰ we think it makes sense to focus GAN research on tasks where this is fine or even helpful. GANs are likely to be well-suited to tasks with a perceptual flavor. Graphics applications like image synthesis, image translation, image infilling, and attribute manipulation all fall under this umbrella.

How should we evaluate GANs on these perceptual tasks? Ideally, we would just use a human judge, but this is expensive. A cheap proxy is to see if a classifier can distinguish between real and fake examples. This is called a classifier two-sample test (C2STs) [68, 69, 60]. The main issue with C2STs is that if the Generator has even a minor defect that's systematic across samples (e.g., [70]) this will dominate the evaluation.

Ideally, we'd have a holistic evaluation that isn't dominated by a single factor. One approach might be to make a critic that is blind to the dominant defect. But once we do this, some other defect may dominate, requiring a new critic, and so on. If we do this iteratively, we could get a kind of 'Gram-Schmidt procedure for critics', creating an ordered list of the most important defects and critics that ignore them. Perhaps this can be done by performing PCA on the critic activations [71] and progressively throwing out more and more of the higher variance components.

Finally, we could evaluate on humans despite the expense. This would allow us to measure the thing that we actually care about. This kind of approach can be made less expensive by predicting human answers and only interacting with a real human when the prediction is uncertain [72].

How does GAN Training Scale with Batch Size?

Large minibatches have helped to scale up image classification [73, 74, 75, 76, 77] — can they also help us scale up GANs? Large minibatches may be especially important for effectively using highly parallel hardware accelerators [78, 79].

At first glance, it seems like the answer should be yes — after all, the discriminator in most GANs is just an image classifier. Larger batches can accelerate training if it is bottlenecked on gradient noise. However, GANs have a separate bottleneck that classifiers don't: the training procedure can diverge. Thus, we can state our problem:

- Problem 6 How does GAN training scale with batch size?
How big a role does gradient noise play in GAN training?
Can GAN training be modified so that it scales better with batch size?

There's some evidence that increasing minibatch size improves quantitative results and reduces training time [4]. If this phenomenon is robust, it would suggest that gradient noise is a dominating factor. However, this hasn't been systematically studied, so we believe this question remains open.

Can alternate training procedures make better use of large batches? Optimal Transport GANs [80] theoretically have better convergence properties than normal GANs, but need a large batch size because they try to align batches of samples and training data. As a result, they seem like a promising candidate for scaling to very large batch sizes.

Finally, asynchronous SGD [81, 82, 83, 84] could be a good alternative for making use of new hardware. In this setting, the limiting factor tends to be that gradient updates are computed on 'stale' copies of the parameters. But GANs seem to actually benefit from training on past parameter snapshots [56, 85], so we might ask if asynchronous SGD interacts in a special way with GAN training.

What is the Relationship Between GANs and Adversarial Examples?

It's well known [86] that image classifiers suffer from adversarial examples: human-imperceptible perturbations that cause classifiers to give the wrong output when added to images. It's also now known that there are classification problems which can normally be efficiently learned, but are exponentially hard to learn robustly [87, 88].

Since the GAN discriminator is an image classifier, one might worry about it suffering from adversarial examples. Despite the large bodies of literature on GANs and adversarial examples, there doesn't seem to be much work on how they relate.¹¹ Thus, we can ask the question:

- Problem 7 How does the adversarial robustness of the discriminator affect GAN training?

How can we begin to think about this problem? Consider a fixed discriminator **D**. An adversarial example for **D** would exist if there were a generator sample **G(z)** correctly classified as fake and a small perturbation **p** such that **G(z) + p** is classified as real. With a GAN, the concern would be that the gradient update for the generator would yield a new generator **G'** where **G'(z) = G(z) + p**.

Is this concern realistic? [89] shows that deliberate attacks on generative models can work, but we are more worried about something you might call an ‘accidental attack’. There are reasons to believe that these accidental attacks are less likely. First, the generator is only allowed to make one gradient update before the discriminator is updated again. In contrast, current adversarial attacks are typically run for tens of iterations [90]. Second, the generator is optimized given a batch of samples from the prior, and this batch is different for every gradient step. Finally, the optimization takes place in the space of parameters of the generator rather than in pixel space. However, none of these arguments decisively rules out the generator creating adversarial examples. We think this is a fruitful topic for further exploration.

Acknowledgments

We would like to thank Colin Raffel, Ben Poole, Eric Jang, Dustin Tran, Alex Kurakin, David Berthelot, Aurko Roy, Ian Goodfellow, and Matt Hoffman for helpful discussions and feedback. We would especially like to single out Chris Olah, who provided substantial feedback on the text and help with editing.

Footnotes

1. We also believe that writing this article has clarified our thinking about GANs, and we would encourage other researchers to write similar articles about their own sub-fields. [↩]
2. This statement shouldn’t be taken too literally. Those are useful terms for describing fuzzy clusters in ‘model-space’, but there are models that aren’t easy to describe as belonging to just one of those clusters. I’ve also left out VAEs [10] entirely; they’re arguably no longer considered state-of-the-art at any tasks of record. [↩]
3. Even in this case, there might still be other reasons to use adversarial training in contexts like image-to-image translation. It also might still make sense to combine adversarial training with maximum-likelihood training. [↩]
4. For instance, it’s possible that the progressive growing technique could be applied to Flow Models as well. [↩]
5. Parallelizable is somewhat imprecise in this context. We mean that sampling from Flow Models must in general be done sequentially, one observation at a time. There may be ways around this limitation though [21]. [↩]
6. These assumptions are very strict. The referenced paper assumes (roughly speaking) that the equilibrium we are looking for exists and that we are already very close to it. [↩]
7. Among other things, it’s assumed that we can first learn the means of the Gaussian and then learn the variances. [↩]
8. A fact that practitioners already kind of suspected. [↩]
9. The generator being a flow model allows for computation of exact log-likelihoods in this case. [↩]
10. Though trying to fix this issue is a valid research agenda as well. [↩]
11. There is work on using GANs to generate adversarial examples, but this is not quite the same thing. [↩]

References

1. Conditional Image Synthesis With Auxiliary Classifier GANs [\[PDF\]](#)
Odena, A., Olah, C. and Shlens, J., 2016. ArXiv e-prints.
2. Self-Attention Generative Adversarial Networks [\[PDF\]](#)
Zhang, H., Goodfellow, I., Metaxas, D. and Odena, A., 2018. ArXiv e-prints.
3. Spectral Normalization for Generative Adversarial Networks [\[PDF\]](#)
Miyato, T., Kataoka, T., Koyama, M. and Yoshida, Y., 2018. CoRR, Vol abs/1802.05957.
4. Large Scale GAN Training for High Fidelity Natural Image Synthesis [\[PDF\]](#)
Brock, A., Donahue, J. and Simonyan, K., 2018. ArXiv e-prints.
5. A style-based generator architecture for generative adversarial networks [\[PDF\]](#)
Karras, T., Laine, S. and Aila, T., 2018. arXiv preprint arXiv:1812.04948.
6. Open Questions in Physics [\[HTML\]](#)
John, B., 2010.
7. Not especially famous, long-open problems which anyone can understand [\[link\]](#)
Overflow, S., 2012.
8. Hilbert's Problems [\[link\]](#).
Hilbert, D., 1900.
9. Smale's Problems [\[link\]](#)
Smale, S., 1998.
10. Auto-Encoding Variational Bayes [\[PDF\]](#)
Kingma, D.P. and Welling, M., 2013. arXiv preprint arXiv:1312.6114.
11. NICE: Non-linear Independent Components Estimation [\[PDF\]](#)
Dinh, L., Krueger, D. and Bengio, Y., 2014. CoRR, Vol abs/1410.8516.
12. Density estimation using Real NVP [\[PDF\]](#)
Dinh, L., Sohl-Dickstein, J. and Bengio, S., 2016. CoRR, Vol abs/1605.08803.
13. Glow: Generative Flow with Invertible 1x1 Convolutions [\[PDF\]](#)
Kingma, D. and Dhariwal, P., 2018. ArXiv e-prints.
14. Normalizing Flows Tutorial [\[HTML\]](#)
Jang, E., 2016.
15. Pixel Recurrent Neural Networks [\[PDF\]](#)
Oord, A.v.d., Kalchbrenner, N. and Kavukcuoglu, K., 2016. CoRR, Vol abs/1601.06759.
16. Conditional Image Generation with PixelCNN Decoders [\[PDF\]](#)
Oord, A.v.d., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A. and Kavukcuoglu, K., 2016. CoRR, Vol abs/1606.05328.
17. PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications [\[PDF\]](#)
Salimans, T., Karpathy, A., Chen, X. and Kingma, D.P., 2017. CoRR, Vol abs/1701.05517.
18. WaveNet: A Generative Model for Raw Audio [\[PDF\]](#)
Oord, A.v.d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A.W. and Kavukcuoglu, K., 2016. CoRR, Vol abs/1609.03499.
19. Progressive Growing of GANs for Improved Quality, Stability, and Variation [\[PDF\]](#)
Karras, T., Aila, T., Laine, S. and Lehtinen, J., 2017. CoRR, Vol abs/1710.10196.

20. Variational Inference with Normalizing Flows [\[PDF\]](#)
Jimenez Rezende, D. and Mohamed, S., 2015. ArXiv e-prints.
21. Parallel WaveNet: Fast High-Fidelity Speech Synthesis [\[PDF\]](#)
Oord, A.v.d., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., Driessche, G.v.d., Lockhart, E., Cobo, L.C., Stimberg, F., Casagrande, N., Grewe, D., Noury, S., Dieleman, S., Elsen, E., Kalchbrenner, N., Zen, H., Graves, A., King, H., Walters, T., Belov, D. and Hassabis, D., 2017. CoRR, Vol abs/1711.10433.
22. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services
Gilbert, S. and Lynch, N., 2002. Acm Sigact News, Vol 33(2), pp. 51--59. ACM.
23. Flow-GAN: Bridging implicit and prescribed learning in generative models [\[PDF\]](#)
Grover, A., Dhar, M. and Ermon, S., 2017. CoRR, Vol abs/1705.08868.
24. Comparison of maximum likelihood and gan-based training of real nvp's [\[PDF\]](#)
Danilhelka, I., Lakshminarayanan, B., Uria, B., Wierstra, D. and Dayan, P., 2017. arXiv preprint arXiv:1705.05263.
25. Is Generator Conditioning Causally Related to GAN Performance? [\[PDF\]](#)
Odena, A., Buckman, J., Olsson, C., Brown, T.B., Olah, C., Raffel, C. and Goodfellow, I., 2018. arXiv preprint arXiv:1802.08768.
26. Gradient-Based Learning Applied to Document Recognition [\[PDF\]](#)
LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Proceedings of the IEEE.
27. Learning Multiple Layers of Features from Tiny Images [\[PDF\]](#)
Krizhevsky, A., 2009.
28. An analysis of single-layer networks in unsupervised feature learning [\[PDF\]](#)
Coates, A., Ng, A. and Lee, H., 2011. Proceedings of the fourteenth international conference on artificial intelligence and statistics, pp. 215--223.
29. Deep Learning Face Attributes in the Wild [\[PDF\]](#)
Liu, Z., Luo, P., Wang, X. and Tang, X., 2015. Proceedings of International Conference on Computer Vision (ICCV).
30. ImageNet Large Scale Visual Recognition Challenge [\[PDF\]](#)
Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C. and Fei-Fei, L., 2015. International Journal of Computer Vision (IJCV), Vol 115(3), pp. 211-252. DOI: 10.1007/s11263-015-0816-y
31. PSA [\[link\]](#)
Raffel, C., 2018.
32. Are GANs Created Equal? A Large-Scale Study [\[PDF\]](#)
Lucic, M., Kurach, K., Michalski, M., Gelly, S. and Bousquet, O., 2017. ArXiv e-prints.
33. Disconnected Manifold Learning for Generative Adversarial Networks [\[PDF\]](#)
Khayatkhoei, M., Singh, M. and Elgammal, A., 2018. ArXiv e-prints.
34. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks [\[PDF\]](#)
Zhu, J., Park, T., Isola, P. and Efros, A.A., 2017. CoRR, Vol abs/1703.10593.
35. Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks [\[PDF\]](#)
Bousmalis, K., Silberman, N., Dohan, D., Erhan, D. and Krishnan, D., 2016. CoRR, Vol abs/1612.05424.
36. Improved training of wasserstein gans [\[PDF\]](#)
Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. and Courville, A.C., 2017. Advances in Neural Information Processing Systems, pp. 5767--5777.
37. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient [\[PDF\]](#)
Yu, L., Zhang, W., Wang, J. and Yu, Y., 2016. CoRR, Vol abs/1609.05473.

38. MaskGAN: Better Text Generation via Filling in the ____ [PDF]
Fedus, W., Goodfellow, I. and Dai, A., 2018. ArXiv e-prints.
39. Long short-term memory
Hochreiter, S. and Schmidhuber, J., 1997. Neural computation, Vol 9(8), pp. 1735--1780. MIT Press.
40. Geometric deep learning: going beyond Euclidean data [PDF]
Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A. and Vandergheynst, P., 2016. CoRR, Vol abs/1611.08097.
41. NetGAN: Generating Graphs via Random Walks [PDF]
Bojchevski, A., Shchur, O., Zugner, D. and Gunnemann, S., 2018. ArXiv e-prints.
42. Synthesizing Audio with Generative Adversarial Networks [PDF]
Donahue, C., McAuley, J. and Puckette, M., 2018. CoRR, Vol abs/1802.04208.
43. GANSynth: Adversarial Neural Audio Synthesis [link]
Engel, J., Agrawal, K.K., Chen, S., Gulrajani, I., Donahue, C. and Roberts, A., 2019. International Conference on Learning Representations.
44. OpenAI Five [link]
OpenAI,, 2018.
45. Gradient descent GAN optimization is locally stable [PDF]
Nagarajan, V. and Kolter, J.Z., 2017. Advances in Neural Information Processing Systems, pp. 5585--5595.
46. Which Training Methods for GANs do actually Converge? [HTML]
Mescheder, L., Geiger, A. and Nowozin, S., 2018. Proceedings of the 35th International Conference on Machine Learning, Vol 80, pp. 3481--3490. PMLR.
47. Understanding GANs: the LQG Setting [link]
Feizi, S., Suh, C., Xia, F. and Tse, D., 2018.
48. Global Convergence to the Equilibrium of GANs using Variational Inequalities [PDF]
Gemp, I. and Mahadevan, S., 2018. ArXiv e-prints.
49. The Mechanics of n-Player Differentiable Games [PDF]
Balduzzi, D., Racaniere, S., Martens, J., Foerster, J., Tuyls, K. and Graepel, T., 2018. CoRR, Vol abs/1802.05642.
50. Approximation and Convergence Properties of Generative Adversarial Learning [PDF]
Liu, S., Bousquet, O. and Chaudhuri, K., 2017. CoRR, Vol abs/1705.08991.
51. The Inductive Bias of Restricted f-GANs [PDF]
Liu, S. and Chaudhuri, K., 2018. CoRR, Vol abs/1809.04542.
52. On the Limitations of First-Order Approximation in GAN Dynamics [PDF]
Li, J., Madry, A., Peebles, J. and Schmidt, L., 2017. ArXiv e-prints.
53. The Loss Surface of Multilayer Networks [PDF]
Choromanska, A., Henaff, M., Mathieu, M., Arous, G.B. and LeCun, Y., 2014. CoRR, Vol abs/1412.0233.
54. GANGs: Generative Adversarial Network Games [PDF]
Oliehoek, F.A., Savani, R., Gallego-Posada, J., Van der Pol, E., De Jong, E.D. and Gros, R., 2017. arXiv preprint arXiv:1712.00679.
55. Beyond Local Nash Equilibria for Adversarial Networks [PDF]
Oliehoek, F., Savani, R., Gallego, J., van der Pol, E. and Gross, R., 2018. ArXiv e-prints.
56. An Online Learning Approach to Generative Adversarial Networks [PDF]
Grnarova, P., Levy, K.Y., Lucchi, A., Hofmann, T. and Krause, A., 2017. CoRR, Vol abs/1706.03269.

57. Improved Techniques for Training GANs [\[PDF\]](#)
Salimans, T., Goodfellow, I.J., Zaremba, W., Cheung, V., Radford, A. and Chen, X., 2016. CoRR, Vol abs/1606.03498.
58. GANs Trained by a Two Time-Scale Update Rule Converge to a Nash Equilibrium [\[PDF\]](#)
Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G. and Hochreiter, S., 2017. CoRR, Vol abs/1706.08500.
59. A Note on the Inception Score [\[PDF\]](#)
Barratt, S. and Sharma, R., 2018. arXiv preprint arXiv:1801.01973.
60. Towards GAN Benchmarks Which Require Generalization [\[link\]](#)
Gulrajani, I., Raffel, C. and Metz, L., 2019. International Conference on Learning Representations.
61. Multiscale structural similarity for image quality assessment
Wang, Z., Simoncelli, E.P. and Bovik, A.C., 2003. The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, Vol 2, pp. 1398–1402.
62. On the Quantitative Analysis of Decoder-Based Generative Models [\[PDF\]](#)
Wu, Y., Burda, Y., Salakhutdinov, R. and Grosse, R.B., 2016. CoRR, Vol abs/1611.04273.
63. Annealed importance sampling
Neal, R.M., 2001. Statistics and computing, Vol 11(2), pp. 125--139. Springer.
64. Geometry Score: A Method For Comparing Generative Adversarial Networks [\[PDF\]](#)
Khrulkov, V. and Oseledets, I.V., 2018. CoRR, Vol abs/1802.02664.
65. Assessing Generative Models via Precision and Recall [\[PDF\]](#)
Sajjadi, M.S., Bachem, O., Lucic, M., Bousquet, O. and Gelly, S., 2018. arXiv preprint arXiv:1806.00035.
66. Skill Rating for Generative Models [\[PDF\]](#)
Olsson, C., Bhupatiraju, S., Brown, T., Odena, A. and Goodfellow, I., 2018. ArXiv e-prints.
67. Discriminator Rejection Sampling [\[PDF\]](#)
Azadi, S., Olsson, C., Darrell, T., Goodfellow, I. and Odena, A., 2018. ArXiv e-prints.
68. Revisiting Classifier Two-Sample Tests [\[PDF\]](#)
Lopez-Paz, D. and Oquab, M., 2016. ArXiv e-prints.
69. Parametric Adversarial Divergences are Good Task Losses for Generative Modeling [\[PDF\]](#)
Huang, G., Berard, H., Touati, A., Gidel, G., Vincent, P. and Lacoste-Julien, S., 2017. arXiv preprint arXiv:1708.02511.
70. Deconvolution and Checkerboard Artifacts [\[link\]](#)
Odena, A., Dumoulin, V. and Olah, C., 2016. Distill. DOI: 10.23915/distill.00003
71. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability
Raghu, M., Gilmer, J., Yosinski, J. and Sohl-Dickstein, J., 2017. Advances in Neural Information Processing Systems, pp. 6076--6085.
72. Deep reinforcement learning from human preferences
Christian, P.F., Leike, J., Brown, T., Martic, M., Legg, S. and Amodei, D., 2017. Advances in Neural Information Processing Systems, pp. 4299--4307.
73. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour [\[PDF\]](#)
Goyal, P., Dollar, P., Girshick, R.B., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y. and He, K., 2017. CoRR, Vol abs/1706.02677.
74. Scaling sgd batch size to 32k for imagenet training [\[PDF\]](#)
You, Y., Gitman, I. and Ginsburg, B., 2017.

75. Train longer, generalize better: closing the generalization gap in large batch training of neural networks [\[PDF\]](#)
Hoffer, E., Hubara, I. and Soudry, D., 2017. Advances in Neural Information Processing Systems 30, pp. 1731--1741. Curran Associates, Inc.
76. Don't Decay the Learning Rate, Increase the Batch Size [\[PDF\]](#)
Smith, S.L., Kindermans, P. and Le, Q.V., 2017. CoRR, Vol abs/1711.00489.
77. An Empirical Model of Large-Batch Training
McCandlish, S., Kaplan, J., Amodei, D. and Team, O.D., 2018. arXiv e-prints.
78. Science and research policy at the end of Moore's law [\[link\]](#)
Khan, H.N., Hounshell, D.A. and Fuchs, E.R., 2018. Nature Electronics, Vol 1(1), pp. 14. Nature Publishing Group.
79. In-datacenter performance analysis of a tensor processing unit [\[PDF\]](#)
Jouppi, N.P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A. and others,, 2017. Computer Architecture (ISCA), 2017 ACM/IEEE 44th Annual International Symposium on, pp. 1--12.
80. Improving GANs using optimal transport [\[PDF\]](#)
Salimans, T., Zhang, H., Radford, A. and Metaxas, D., 2018. arXiv preprint arXiv:1803.05573.
81. Large Scale Distributed Deep Networks [\[PDF\]](#)
Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., Le, Q.V. and Ng, A.Y., 2012. Advances in Neural Information Processing Systems 25, pp. 1223--1231. Curran Associates, Inc.
82. Deep learning with Elastic Averaging SGD [\[PDF\]](#)
Zhang, S., Choromanska, A. and LeCun, Y., 2014. CoRR, Vol abs/1412.6651.
83. Staleness-aware Async-SGD for Distributed Deep Learning [\[PDF\]](#)
Zhang, W., Gupta, S., Lian, X. and Liu, J., 2015. CoRR, Vol abs/1511.05950.
84. Faster Asynchronous SGD [\[PDF\]](#)
Odena, A., 2016. ArXiv e-prints.
85. The Unusual Effectiveness of Averaging in GAN Training [\[PDF\]](#)
Yazici, Y., Foo, C., Winkler, S., Yap, K., Piliouras, G. and Chandrasekhar, V., 2018. ArXiv e-prints.
86. Intriguing properties of neural networks [\[PDF\]](#)
Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. and Fergus, R., 2013. arXiv preprint arXiv:1312.6199.
87. Adversarial examples from computational constraints [\[PDF\]](#)
Bubeck, S., Price, E. and Razenshteyn, I., 2018. ArXiv e-prints.
88. Efficient noise-tolerant learning from statistical queries
Kearns, M., 1998. Journal of the ACM (JACM), Vol 45(6), pp. 983--1006. ACM.
89. Adversarial examples for generative models [\[PDF\]](#)
Kos, J., Fischer, I. and Song, D., 2018. 2018 IEEE Security and Privacy Workshops (SPW), pp. 36--42.
90. Towards deep learning models resistant to adversarial attacks [\[PDF\]](#)
Madry, A., Makelov, A., Schmidt, L., Tsipras, D. and Vladu, A., 2017. arXiv preprint arXiv:1706.06083.

Updates and Corrections

If you see mistakes or want to suggest changes, please [create an issue on GitHub](#).

Reuse

Diagrams and text are licensed under Creative Commons Attribution [CC-BY 4.0](#) with the source available on [GitHub](#), unless noted otherwise. The figures that have been reused from other sources don't fall under this license and can be recognized by a note in their caption: "Figure from ...".

Citation

For attribution in academic contexts, please cite this work as

Odena, "Open Questions about Generative Adversarial Networks", Distill, 2019.

BibTeX citation

```
@article{odena2019open,
  author = {Odena, Augustus},
  title = {Open Questions about Generative Adversarial Networks},
  journal = {Distill},
  year = {2019},
  note = {\url{https://distill.pub/2019/gan-open-problems}},
  doi = {10.23915/distill.00018}
}
```

 Distill is dedicated to clear explanations of machine learning

[About](#) [Submit](#) [Prize](#) [Archive](#) [RSS](#) [GitHub](#) [Twitter](#) ISSN 2476-0757