Learning Implicit Generative Models with Method of Learned Moments

Suman Ravuri, Shakir Mohamed, Mihaela Rosca, Oriol Vinyals
DeepMind
London, N1C 4AG, UK
{ravuris,shakir,mihaelacr,vinyals}@google.com

Abstract

We introduce a method of moments (MoM) framework for training implicit generative models. Previous attempts at using MoM-like objectives have typically encountered two problems: it is often difficult to define the tens of millions of moments needed to learn the generator parameters, and it is hard to determine which properties are useful when creating moments. To address this, we first define our millions of moments $\Phi(X)$ as the gradient of the output of a discriminator with respect to its parameters. Second, we use asymptotic theory to highlight desiderata for moments – namely they should minimize the asymptotic variance of estimated generator parameters – and introduce an objective to learn moments. The objective created by this "Method of Learned Moments" (MoLM) is far more stationary than GAN alternatives, requiring updates of the discriminator only every 2,000 generator steps. On image generation, we show that MoLM improves sample quality for CelebA, Color MNIST, and CIFAR-10, and results are competitive with adversarial methods on CelebA and Color MNIST.

1 Introduction and Related Work

Data such as images, speech, or music often arise from complicated distributions. The gold standard for modeling such distributions are likelihoods – as likelihoods exhibit nice theoretic properties – but estimating them accurately is both intellectually and computationally challenging. For researchers interested in generating more realistic samples, efforts have shifted to developing alternative losses. Neural networks that estimate integral probability metrics [1, 19] or divergences [4, 11, 15] now lead to objectives that can train implicit generative models. On image generation tasks, these neural samplers produce more visually appealing samples. Perhaps crucially, these alternative criteria share an "adversarial" training strategy.

Other work on learning neural generators using an "infinite moment"-matching method [2, 9], an application of kernel mean embedding of distributions [14, 18], has so far produced worse results. That sample quality only improved when combined with adversarial training [8] suggests that the adversarial methods are the "secret sauce" for creating high-quality image generation models. As a result, many research programs have currently focused on fixing issues within the adversarial paradigm.

This paper posits an alternative hypothesis: the relatively poor performance of these early moment-matching methods was the result of poor design of moments, and better choices of moments can yield much higher-quality samples. We appeal to asymptotic theory to determine desiderata for moments (namely that they minimize the asymptotic variance of estimated generator parameters) and explicitly specify and learn moments to train the generator. It is this theory that is used in the literature of "generalized method of moments" in econometrics to reweight moments to make estimators more statistically efficient [5, 6].

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

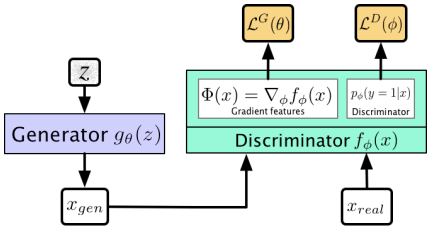

Figure 1: Illustration of method of learned moments architecture.

We make two contributions in this work:

- First, to specify the millions of moments needed to train neural generators – intuitively at least one per generator parameter – we introduce a neural network architecture akin to a discriminator, whose gradient with respect to its parameters defines the moments.
- Second, we use asymptotic theory to create an objective that learns moments, which results in higher sample quality.

The resulting algorithm, denoted the "Method of Learned Moments" (MoLM), is substantially more stationary than the adversarial algorithms. In particular, we require discriminator updates only every 2,000 generator steps rather than every step. Furthermore, sample quality rivals that of adversarial methods while exhibiting no mode collapse. Before we describe the method in Sections 2.2 and 2.3, we provide a short reintroduction to the method of moments (MoM) that will also serve as motivation for our method.

2 Method

2.1 Method of Moments Background

Suppose our data are drawn i.i.d. from $X_i \sim P$, and our samples S are drawn i.i.d. from a model p_γ, whose parameters $\gamma \in \mathbb{R}^n$ we wish to learn. MoM estimation requires us to define feature functions $\Phi(x) \in \mathbb{R}^k$ and estimate γ such that $1/N \sum_{i=1}^N \Phi(X_i) = \mathbb{E}_{S\sim p_\gamma}[\Phi(S)]$. If $p_\gamma \overset{d}{=} P$ for some γ, under certain regularity conditions on Φ (defined in Appendix B), this feature matching will yield a consistent estimator of γ. When we have access to the likelihood, we can recover the maximum likelihood estimate by setting $\Phi(x) = \nabla_\gamma \log p_\gamma(x)$ and noting that the expected value of the score function is zero at γ^*. Since maximum likelihood estimators are asymptotically efficient, they are generally preferable to their moment counterparts.

In the implicit generative model setting in which a parametric sampler g_θ replaces the model, lacking access to p_γ precludes the use of maximum likelihood. MoM estimation, however, is still applicable by replacing $\mathbb{E}_{S\sim p_\gamma}[\Phi(S)]$ with $\mathbb{E}_{Z\sim\mathcal{N}(0,1)}[\Phi(g_\theta(Z))]$, where Z is a draw from a noise distribution such as Gaussian or Uniform noise. If $g_\theta(Z)$ is sufficiently expressive to model the data distribution, the same regularity conditions on Φ will ensure a consistent estimator of generator parameters θ. While consistency is guaranteed, the asymptotic efficiency argument of maximum likelihood implies that the quality of moments $\Phi(X)$ matter. To obtain better moments, we will explicitly create them (rather than have them implicitly defined by choice of kernel function) and optimize those moments to be more statistically efficient. This approach introduces two hurdles not present in previous work: 1) defining millions (and often tens of millions) of sufficiently different moments and 2) creating an objective to learn desirable moments. How we solve these two issues comprises our contribution.

2.2 Defining Moments

Explicitly specifying moments for neural samplers seems especially daunting, since generally one needs at least as many moments as sampler parameters for moment matching to yield a consistent estimator of generator parameters. For the DCGAN [16] architecture with batch normalization, this requires around 20M moments. Our solution is to create a neural network $f_\phi(X)$ akin to the discriminator of GANs, and define moments as:

$$\Phi(X) = \nabla_\phi f_\phi(X)$$

The objective is the squared error between moments of the data and samples:

$$\mathcal{L}^G(\theta) = \frac{1}{2}\|\mathbb{E}_{X\sim P}[\nabla_\phi f_\phi(X)] - \mathbb{E}_{Z\sim\mathcal{N}(0,1)}[\nabla_\phi f_\phi(g_\theta(Z))]\|_2^2 \tag{1}$$

Figure 2.2 illustrates our setup. In practice, we take Monte Carlo estimates of both expectations. Unlike adversarial methods, the moment-matching objective is stationary. As a result, we can reduce the variance of $\mathbb{E}_{X\sim P}[\nabla_\phi f_\phi(X)]$ by averaging over the entire dataset.

One may think that moments $\nabla_\phi f_\phi(X)$ for random ϕ are sufficiently good to train neural samplers. Unfortunately, as we show in Section 3, this is not the case, as random moments are not sufficiently discriminative. In the next subsection, we will connect discriminativeness of moments with the asymptotic variance of the moment estimator and create an objective that learns better moments.

2.3 Learning Efficient Moments

One desirable aspect of the moment-matching framework is a developed asymptotic theory that provides large-sample behavior of the moment estimator. Intuitively, it tells us that our estimated generator parameters after N datapoints is roughly distributed as $\hat{\theta}_N \dot{\sim} \mathcal{N}(\theta^*, {}^V\!/_N)$, where V is the asymptotic variance. Minimizing V makes estimation of generator parameters more data-efficient and depends on the quality of $\Phi(X)$.

Assuming consistency of the estimator, standard results state the following:

Theorem 1. *Let $e(\theta) = \mathbb{E}[\Phi(g_\theta(Z))]$ be a one-to-one function on an open set $\Theta \subset \mathbb{R}^d$ and continuously differentiable at θ^* with nonsingular Jacobian $G = J_{\theta^*} e(\theta)$. Then assuming $\mathbb{E}[\|\Phi(g_{\theta^*}(Z))\|^2] < \infty$, moment estimators $\hat{\theta}_N$ exist and satisfy*

$$\sqrt{N}(\hat{\theta}_N - \theta^*) \to \mathcal{N}(0, (G^T)^{-1}\Sigma G^{-1}), \qquad \Sigma \coloneqq \mathbb{E}_Z[\Phi(g_{\theta^*}(Z))\Phi(g_{\theta^*}(Z))^T]$$

Proof. See Theorem 4.1 in [21] □

Minimizing this asymptotic covariance, known as the inverse Godambe Information Matrix [3] when $\Sigma \coloneqq cov(\Phi(g_{\theta^*}(Z)))$, requires balancing G and Σ. G asserts that one should maximize the difference in features between the optimal generator parameters and those a small distance away, while Σ expresses that one should minimize the second moment of the features for the optimal generator parameters. Note that although the same tradeoff applies to the objective defined in Equation 1, its asymptotic variance is somewhat more complicated. Due to space constraints, that variance is defined in Appendix B.

Since we do not have access to θ^*, exact computation of G and Σ is impossible. Under the assumption that the generator is sufficiently expressive to represent the data distribution, then $\Phi(X) = \Phi(g_{\theta^*}(Z))$ and $\Sigma = \mathbb{E}_X[\Phi(X)\Phi(X)^T]$. For G, by definition of the Jacobian:

$$G(\theta - \theta^*) = \mathbb{E}_Z[\Phi(g_\theta(Z))] - \mathbb{E}_X[\Phi(X)] + o(\|\theta - \theta^*\|)$$

For θ near the optimum, maximizing the difference in expected features also maximizes the "directional Jacobian" G. In early experiments more directly optimizing this difference, moments became collinear and as a result the estimator of θ was no longer consistent. Thus, we make a second approximation. Let $D_\phi(X) = P(y = 1|X)$ be the probability that X is a real image and $\Phi(X) = \nabla_\phi f_\phi(X) \coloneqq \nabla_\phi \log {}^{P(y=1|X)}\!/_{P(y=-1|X)}$, and constrain ϕ such that $0 < D_\phi(X) < 1 \;\; \forall X \in \mathcal{X}$. Since the expected value of the score function $\mathbb{E}_Y[\nabla_\phi f_\phi(Y)] \approx 0$ when the binary classifier is trained on data and samples (Y being an equally weighted mixture distribution of data and samples), the moments provide some measure of separation. When the log-odds ratio is included in the feature, the resulting kernel $K(x, y) = \Phi(x)^T\Phi(y)$ is known as Tangent of Posterior Odds Kernel [20].

To control Σ and ensure $0 < D_\phi(X) < 1$, we add a quadratic penalty on the squared norm of gradient, averaged over the data, to be close to 1. The discriminator objective is now:

$$\mathcal{L}^D(\phi) = \mathbb{E}_X[\log D_\phi(X)] + \mathbb{E}_Z[\log(1 - D_\phi(g_\theta(Z)))] + \lambda(\frac{\mathbb{E}_X[\|\Phi(X)\|^2]}{k} - 1)^2 \qquad \Phi(X) \in \mathbb{R}^k \tag{2}$$

The upshot of defining gradients as moments is that each parameterization of f_ϕ, subject to regularity conditions, produces a consistent estimator of θ. On the other hand, each set of moments is not necessarily asymptotically efficient. So, we learn better Φ iteratively. This leads to more stationary losses since we only update the discriminator parameters for 100 steps every 2,000th update of the generator. Algorithm 1 describes the proposed method. (Please see Appendix A for hyperparameters).

Algorithm 1: Method of Learned Moments

Input: Learning rate α; number of training steps N_t; N_c, the step in which the discriminator is trained; number of discriminator training steps N_d; norm penalty parameter λ

Initialize generator and discriminator parameters θ and γ respectively ;

for $n = 0, \ldots, N_t$ **do**

  if $n \mod N_c == 0$ **then**

    for $n = 1, \ldots, N_d$ **do**

      $d_\phi \leftarrow \nabla_\phi \mathcal{L}^D(\phi)$;

      $\phi_{t+1} \leftarrow \phi_t - \alpha \cdot \text{AdamOptimizer}(\phi_t, d_\phi)$;

    Calculate $\mathbb{E}_X[\Phi(X)]$ over the entire dataset.;

  else

    $d_\theta \leftarrow \nabla_\theta \mathcal{L}^G(\theta)$;

    $\theta_{t+1} \leftarrow \theta_t - \alpha \cdot \text{AdamOptimizer}(\theta_t, d_\theta)$;

3 Experimental Results and Discussion

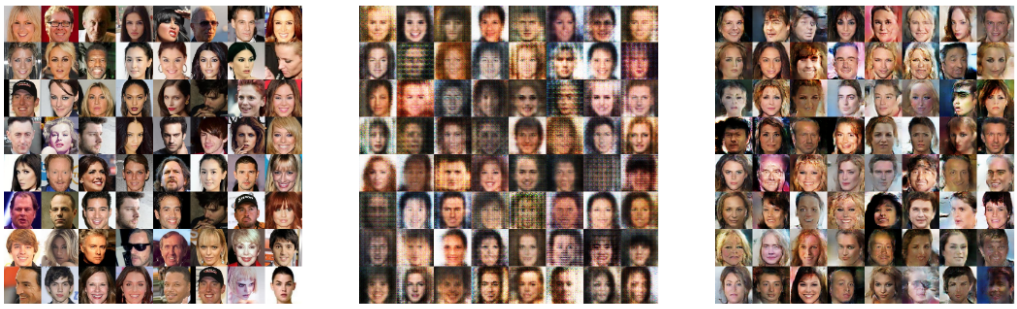

Figure 2: Random CelebA samples. From left to right: 1) data, 2) examples from the generator trained with random discriminator weights, 3) examples from the generator trained with MoLM.

We evaluate our method on three data sets: Color MNIST [12], CelebA [10], and CIFAR-10 [7]. We complement the visual inspection of samples with numerical measures to compare this method to existing work. For CelebA, we use Multi-Scale Structural Similarity (MS-SSIM) [22] to show sample diversity within a single class. Higher scores typically indicate mode collapse, while lower indicate higher diversity and better performance. Numbers lower than the test set may imply underfitting. For CIFAR-10, we include the standard Inception score [17].

Our discriminator is always a convolutional network, and we use the generator used in DCGAN. For details of the specific architectures and hyperparameters used in all our experiments, please see Appendix A in the supplementary material. Note that the models considered here are all unconditional.

Figure 3 highlights the importance of learning moments. While random moments somewhat surprisingly are sufficiently good to learn the structure of faces, learned moments allow the generator to more closely match the data distribution. Moreover, the MS-SSIM for learned features is .371, which is in line with better GAN alternatives and a bit lower than that for the test set: .379. Random features, on the other hand, obtain a score of .411, indicating a higher degree of similarity. Similar results extend to Color MNIST and CIFAR-10. Due to space constraints, Color MNIST and CIFAR-10 samples are relegated to Appendix C.

Finally, as a negative result, while samples and metrics seem competitive with GANs on Color MNIST and CelebA, results on CIFAR-10 are less compelling. The proposed method and random baseline obtains Inception Scores of 5.2 and 3.1, respectively, which lags other GAN baselines that typically score above 6.0. Whether alternative architectures or training methods are needed to improve results on this dataset is left to future work.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

[2] Gintare Karolina Dziugaite, Daniel M Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. *arXiv preprint arXiv:1505.03906*, 2015.

[3] Vidyadhar P Godambe. An optimum property of regular maximum likelihood estimation. *The Annals of Mathematical Statistics*, 31(4):1208–1211, 1960.

[4] 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*, pp. 2672–2680, 2014.

[5] Alastair R Hall. *Generalized method of moments*. Oxford University Press, 2005.

[6] Lars Peter Hansen. Large sample properties of generalized method of moments estimators. *Econometrica: Journal of the Econometric Society*, pp. 1029–1054, 1982.

[7] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[8] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. *arXiv preprint arXiv:1705.08584*, 2017.

[9] Yujia Li, Kevin Swersky, and Richard Zemel. Generative moment matching networks. In *International Conference on Machine Learning*, pp. 1718–1727, 2015.

[10] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[11] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. *arXiv preprint ArXiv:1611.04076*, 2016.

[12] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

[13] Youssef Mroueh and Tom Sercu. Fisher GAN. *CoRR*, abs/1705.09675, 2017. URL `http://arxiv.org/abs/1705.09675`.

[14] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Bernhard Schölkopf, et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

[15] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. *arXiv preprint arXiv:1606.00709*, 2016.

[16] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[17] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. *arXiv preprint arXiv:1606.03498*, 2016.

[18] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In *International Conference on Algorithmic Learning Theory*, pp. 13–31. Springer, 2007.

[19] Dougal J Sutherland, Hsiao-Yu Tung, Heiko Strathmann, Soumyajit De, Aaditya Ramdas, Alex Smola, and Arthur Gretton. Generative models and model criticism via optimized maximum mean discrepancy. *arXiv preprint arXiv:1611.04488*, 2016.

[20] K. Tsuda, M. Kawanabe, G. Rätsch, S. Sonnenburg, and K-R. Müller. A new discriminative kernel from probabilistic models. In *Advances in Neural Information Processing Systems 14*, pp. 977–984, Cambridge, MA, USA, September 2002. Max-Planck-Gesellschaft, MIT Press.

[21] Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 1998.

[22] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on*, volume 2, pp. 1398–1402. IEEE, 2003.

A Achitecture and Hyperparameter Details

As mentioned in the main text, generators for all experiments used the standard DCGAN architecture. The parameters of the generator architectures for different datasets are described in Table 1.

	Color MNIST	CIFAR-10	CelebA
Noise dimension	128	128	256
Projection layer size	4×4×256	4×4×512	4×4×1024
Conv. transpose layer 1 output size	8×8×128	8×8×256	8×8×512
Conv. transpose layer 2 output size	16×16×64	16×16×128	16×16×256
Conv. transpose layer 3 output size	N/A	N/A	32×32×128
Output layer size	32×32×3	32×32×3	64×64×3
Output nonlinearity	sigmoid	tanh	tanh
Kernel size	5×5	4×4	5×5
Batch norm	Yes	Yes	Yes
Number of parameters	1,557,571	3,685,123	20,615,427

Table 1: Generator architectures across different datasets.

Discriminator architectures for Color MNIST and CelebA mirror the generator architectures with one modification: after the last convolutional layer, we replace linear layer of size [4×4×C, 1] with two linear layers of sizes [4×4×C, noise dimension] and [noise dimension, 1], respectively, to ensure that there are at least as many discriminator parameters as generator parameters. For CIFAR-10, we found some improvement using the larger architecture of the semi-supervised learning discriminator used in [13], with the exception that kernel sizes are 4 instead of 3. Please see Appendix F.4 of that paper for more details. Non-linearities between all layers are leaky relus with leaky parameter 0.2. No discriminators use batch normalization.

Table 2 shows the hyperparameters used for all experiments.

	Color MNIST	CIFAR-10	CelebA
Number of training steps N_t	200K	250K	200K
Update discriminator every c-th step N_c	2,000	2,000	2,000
Number of discriminator training steps N_d	100	100	100
Learning rate α	1E-4	1E-4	1E-4
Norm penalty parameter λ	0.1	1.0	1.0
Batch size	1000	200	200

Table 2: Hyperparameters for different datasets.

B Consistency and Asymptotic Normality of Moment Estimators

In this section, we review the consistency and asymptotic normality conditions for moment estimators. Many of these conditions are now standard within a body of literature in econometrics known as "Generalized Method of Moments" and proofs for these conditions can be found in [5]. The consistency conditions for Equation 1 (reproduced below)

$$\mathcal{L}^G(\theta) = \frac{1}{2}\|\mathbb{E}_{X\sim P}[\Phi(X)] - \mathbb{E}_{Z\sim\mathcal{N}(0,1)}[\Phi(g_\theta(Z))]\|_2^2$$

are as follows:

- $\mathbb{E}_{X\sim P}[\Phi(X)] - \mathbb{E}_{Z\sim\mathcal{N}(0,1)}[\Phi(g_\theta(Z))] = 0$ iff $\theta = \theta^*$
- $\Theta \subset \mathbb{R}^k$, $\theta \in \Theta$, Θ is a compact set
- $\mathbb{E}_{X\sim P}[\Phi(X)] - \mathbb{E}_{Z\sim\mathcal{N}(0,1)}[\Phi(g_\theta(Z))]$ is continuous for each θ with probability 1
- $\mathbb{E}_Z[\sup_{\theta\in\Theta} \|\Phi(g_\theta(Z)) - E_X[\Phi(X)]\|] < \infty$

The first condition is typically difficult to verify. A heuristic that seems to work well in practice is to assume that the number of moments is greater than the number of generator parameters, and that the Jacobian of moments with respect to the generator parameters is full-rank. If, in addition, the following conditions are met:

- $\mathbb{E}_{X\sim P}[\Phi(X)] - \mathbb{E}_{Z\sim\mathcal{N}(0,1)}[\Phi(g_\theta(Z))]$ is continuously differentiable in some neighborhood $\mathcal{N}$ around θ^* with probability 1
- $\mathbb{E}_Z[\|\Phi(g_\theta(Z)) - E_X[\Phi(X)]\|^2] < \infty$
- $\mathbb{E}_Z[\sup_{\theta\in\mathcal{N}} \|\nabla_\theta \mathbb{E}_Z[\Phi(g_{\theta^*}(Z))]\|] < \infty$
- $G^T G$ is nonsingular

Then the estimator $\hat{\theta}$ is asymptotically normal with covariance:

$$\sqrt{N}(\hat{\theta} - \theta^*) \to \mathcal{N}(0, (G^T G)^{-1} G^T \Sigma G (G^T G)^{-1})$$

where $\Sigma \coloneqq cov(\Phi(X))$.

For the feature-matching objective defined above, consistency conditions are a bit milder:

- $\mathbb{E}_{X\sim P}[\Phi(X)] - \mathbb{E}_{Z\sim\mathcal{N}(0,1)}[\Phi(g_\theta(Z))] = 0$ iff $\theta = \theta^*$
- $\mathbb{E}_Z[\|\Phi(g_\theta(Z))\|] < \infty$

Again, the first condition is difficult to verify and the heuristic used to show the first consistency condition for Equation 1 also applies here.

C Samples

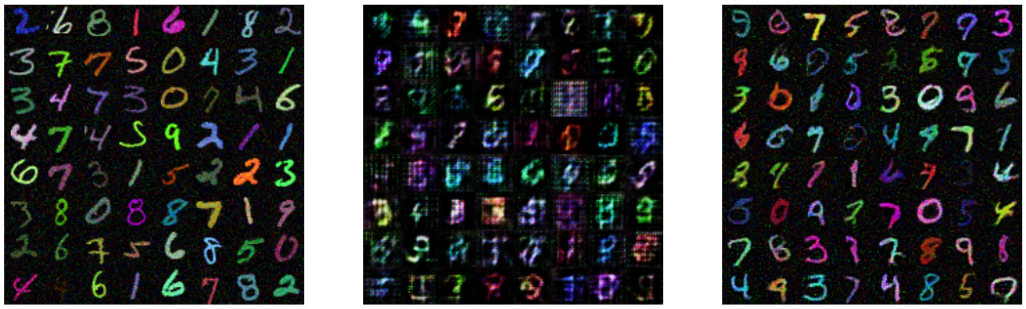

Figure 3: Random Color MNIST samples. From left to right: 1) data, 2) examples from the generator trained with random discriminator weights, 3) examples from the generator trained with MoLM.

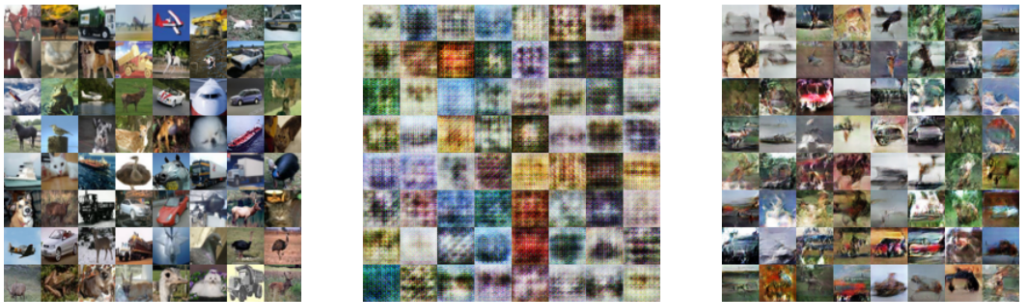

Figure 4: Random CIFAR-10 samples. From left to right: 1) data, 2) examples from the generator trained with random discriminator weights, 3) examples from the generator trained with MoLM.