Supplementary Material for Learning Implicit Generative Models with the Method of Learned Moments

A. Experimental Details

A.1. Experimental Setup

As mentioned in the main text, unless otherwise noted, generators use the standard DCGAN architecture with 4×4 kernels. The structure of the generator architectures for different datasets are described in Table 4.

The moment network for Color MNIST mirror the standard DCGAN discriminator architecture 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 moment network parameters as generator parameters. Furthermore, the generator is trained only with moments from gradient features, as activation features did not improve sample quality. This allowed the use of the Hessian-vector products to more quickly train the generator. Non-linearities between all layers are leaky ReLUs with leaky parameter 0.2.

For CIFAR-10, CelebA, and the daisy portion of ImageNet, we found some improvement using a larger moment network. Again, the moment network mirrors the DCGAN discriminator architecture, but with two changes: prior to each stride-2 convolutional layer we insert a stride-1 layer, and we decrease the kernel size to 3×3. Non-linearities between all layers are leaky ReLUs with leaky parameter 0.2. None of the moment networks use batch normalization. For experiments that used gradient and hidden unit features, hidden units were scaled by a constant factor (known as activation weights in Table 10) since the hidden units had a larger dynamic range than gradient features.

Table 10 shows the hyperparameters used for all experiments with a few exceptions. One is the the stability of MoLM training, which increases the number of objectives from 250 to 400. The second is the comparison of gradient features, activations, and both gradient features and activations, as we vary the size of the moment networks and vary the Adam optimizer's β parameter in that experiment. The last is the comparison with GAN alternatives on CIFAR-10, and the differences are described in the last paragraph of Appendix A.1.

For comparisons, we use two standard, but somewhat flawed metrics: Inception Score (IS), and Fréchet Inception Distance (FID). For IS, we use the standard protocol and calculate scores using 10 batches of 5,000 images (for a total of 50,000) images. For FID, we report distances using 5,000 and 50,000 generated images for comparison with adversarial methods. For all CIFAR-10 experiments in the main text, we use ImageNet-trained networks, as this is the standard network for comparison. As noted by (Rosca et al., 2017; Barratt and Sharma, 2018), however, Inception Scores and Fréchet Inception Distances based on ImageNet-trained networks can lead to misleading results. Therefore, we also include Inception Scores on CIFAR-trained networks[4] in Table 8 for comparison with future work. N.B. we do not include FID results on CIFAR-trained networks, since FID for baseline and proposed methods are extremely low (less than 2.0). We surmise that this is the result of the embedding layer of the CIFAR-trained network being far lower-dimensional than that of the ImageNet-trained one.

On CelebA and CIFAR-10, we tried four GAN variants: GAN (Goodfellow et al., 2014) with and without a gradient penalty (Gulrajani et al., 2017), Wasserstein GAN with a gradient penalty (Gulrajani et al., 2017), and DRAGAN (Kodali et al., 2017) with nonsaturating loss. The same generator architecture was used for the GAN variants as MoLM. The results reported for DRAGAN, GAN-GP, and WGAN-GP were the best obtained in a hyperparameter sweep over discriminator learning rates in 0.0001, 0.0002, 0.0003 and generator learning rates in the same interval. Whenever applicable, the gradient penalty coefficient used was 10. The models were trained using the AdamOptimizer with β_1=0.5 and β_2=0.9. DRAGAN, GAN, and GAN-GP performed one discriminator update per generator update, while WGAN-GP performed 5 discriminator updates for generator updates, for a total of 200,000 generator updates.

On CIFAR-10, we found that our GAN variants had Inception Scores up to 0.2 worse than comparable published results. For completeness, we include these results in Table 7. We did not believe the this would be a reliable indicator of relative performance between adversarial methods and the proposed one. For a more sound comparison, we use GAN-GP and WGAN-GP results from Miyato et al. (2018), as those results are the best we found. It uses a different convolutional generator architecture (its specification can be found in Table 12), which provides the extra benefit of showing that MoLM can train more than just DCGAN gen-

[4]This network can be found at http://download.tensorflow.org/models/frozen_vgg_v1_2018_03_28.tar.gz.

	Color MNIST	CIFAR-10	CelebA	ImageNet Daisy
Noise dimension	128	128	256	256
Projection layer size	4×4×256	4×4×512	4×4×512	4×4×512
Conv. transpose layer 1 output size	8×8×128	8×8×256	8×8×256	8×8×256
Conv. transpose layer 2 output size	16×16×64	16×16×128	16×16×128	16×16×128
Conv. transpose layer 3 output size	N/A	N/A	32×32×64	32×32×64
Conv. transpose layer 4 output size	N/A	N/A	N/A	64×64×32
Output layer size	32×32×3	32×32×3	64×64×3	128×128×3
Output nonlinearity	sigmoid	tanh	tanh	tanh
Hidden nonlinearity	ReLU	ReLU	ReLU	ReLU
Kernel size	5×5	4×4	4×4	4×4
Batch norm	Yes	Yes	Yes	Yes
Number of parameters	1,557,571	3,685,123	4,861,827	4,893,123

Table 4. Generator architectures across different datasets.

	MoLM-512	MoLM-768	MoLM-1024	MoLM-1536
Size-Preserving Layer 1	3×3×3×128	3×3×3×192	3×3×3×256	3×3×3×384
Stride-2 Layer 1	3×3×128×128	3×3×192×192	3×3×256×256	3×3×384×384
Size-Preserving Layer 2	3×3×128×256	3×3×192×384	3×3×256×512	3×3×384×768
Stride-2 Layer 2	3×3×256×256	3×3×384×384	3×3×512×512	3×3×768×768
Size-Preserving Layer 3	3×3×256×512	3×3×384×768	3×3×512×1024	3×3×768×1536
Stride-2 Layer 3	3×3×512×512	3×3×768×768	3×3×1024×1024	3×3×1536×1536
Linear Layer	8,192 ×1	12,288×1	16,384×1	24,576×1
Batch norm	No	No	No	No
Hidden nonlinearity	LReLU	LReLU	LReLU	LReLU
Number of Activations	285,600	420,864	560,128	838,656
Number of Parameters	4,584,577	10,305,217	18,311,425	41,180,545
Number of Total Moments	4,866,177	10,726,081	18,871,553	42,019,201

Table 5. Moment Network Architectures for CIFAR-10

erators. We also believe that those results are among the best for GAN-GP and WGAN-GP for any generator architecture. We also compare to the spectrally-normalized GANs (SN-GAN) in that work. For the DCGAN generator, we compare against MMD-GAN and MMD-RBF as those can be considered moment-based methods. Results were taken from Li et al. (2017). Finally, we include published results for Coulomb GAN (Unterthiner et al., 2017).

A.2. Large Generator Training on CelebA

The experiments in the main text only train generators with up to 5 million parameters. To show the method can scale to a larger number of generator parameters, we doubled the number of channels and increased the kernel size to 5×5. The number of parameters is now 20 million, and Table 11 details the architecture. The moment network mirrors a DCGAN discriminator with 1,024 channels, and adds an extra linear layer to ensure the number of moments is greater than the number of generator parameters. No hidden unit features were used in order to speed up training using the Hessian-vector product trick. Figure 6 shows the result of the experiment: while the generator surprisingly learns some structure of faces using random moments, the generator learns a higher-quality sampler of faces with MoLM.

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 work in econometrics known as "Generalized Method of Moments."

B.1. Consistency and Asymptotic Normality Conditions Squared Error Objective

The consistency and asymptotic normality conditions for the Equation 1 (reproduced below) are taken from (Hall, 2005).

$$\mathcal{L}^G(\theta) = m_N(x_{1,\ldots,N}, \theta)^\intercal W_N m_N(x_{1,\ldots,N}, \theta)$$

We remove the dependence on Φ because it is static. Note that below:

$$m(x, \theta) \coloneqq m_1(x_1, \theta) = \Phi(x) - \mathbb{E}_{p(z)}[\Phi(g_\theta(z))]$$

	CelebA	ImageNet Daisy
Size-Preserving Layer 1	3×3×3×96	3×3×3×48
Stride-2 Layer 1	3×3×96×96	3×3×48×48
Size-Preserving Layer 2	3×3×96×192	3×3×48×96
Stride-2 Layer 2	3×3×192×192	3×3×96×96
Size-Preserving Layer 3	3×3×192×384	3×3×96×192
Stride-2 Layer 1	3×3×384×384	3×3×192×192
Size-Preserving Layer 4	3×3×384×768	3×3×192×384
Stride-2 Layer 2	3×3×768×768	3×3×384×384
Size-Preserving Layer 5	N/A	3×3×384×768
Stride-2 Layer 5	N/A	3×3×768×768
Linear Layer	12,288×1	12,288×1
Batch norm	No	No
Hidden nonlinearity	LReLU	LReLU
Number of Activations	921,600	1,941,504
Number of Parameters	10,551,649	10,612,657
Number of Total Moments	11,473,249	12,554,161

Table 6. Moment Network Architectures for CelebA and ImageNet Daisy

Figure 5. Samples for only activation features, gradient features, and gradient+activation features. Architecture and hyperparameters are using the default generator and MoLM-1024 moment network.

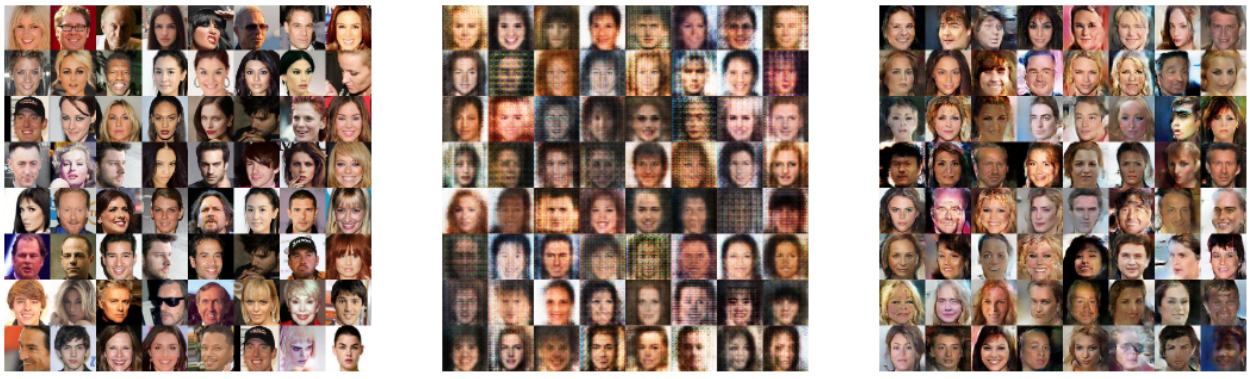

Figure 6. CelebA samples for large generator training. From left to right: 1) data, 2) examples from the generator trained with random moment network weights, 3) examples from the generator trained with MoLM.

Consistency conditions are:

- The $(d \times 1)$ random vectors $\{x_i; \quad i = 1, \dots\}$ form a strictly stationary process with sample space $\mathbf{X} \subset \mathbb{R}^d$.
- The function $m : \mathbf{X} \times \Theta \to \mathbb{R}^k$, where $k < \infty$, satisfies: (i) it is continuous on Θ for each $x_i \in \mathbf{X}$; (ii) $\mathbb{E}_{p(x)}[m(x, \theta)]$ exists and is finite for every $\theta \in \Theta$; (iii)

Table 7. Inception Score for baseline methods and MoLM on CIFAR-10.

Method	Inception Score
GAN	6.75
GAN-GP	6.88
DRAGAN	6.89
WGAN-GP	6.48
MoLM-768	**7.56**

Table 8. Inception Scores using a CIFAR-trained network for MoLM variants.

Architecture	Method	Inception Score
DCGAN	GAN-GP	6.41
DCGAN	WGAN-GP	6.34
DCGAN	DRAGAN	6.35
DCGAN	**MoLM-768**	**6.55**
Conv.	**MoLM-1024**	**6.87**
Conv.	**MoLM-1536**	**7.13**

$\mathbb{E}_{p(x)}[m(x,\theta)]$ is continuous on Θ.

- The random vector X and the parameter vector θ^* satisfy the population moment condition: $\mathbb{E}_{p(x)}[m(x,\theta^*)] = 0$ and $\mathbb{E}_{p(x)}[m(x,\hat{\theta})] \neq 0 \quad \forall \hat{\theta} \neq \theta^*$.
- W_N is a PSD matrix which converges in probability to the PD matrix of constants W.
- The random process $\{X_i, -\infty < i < \infty\}$ is ergodic.
- Θ is a compact set.
- $\mathbb{E}_{p(x)}[\sup_{\theta\in\Theta} \|m(X,\theta)\|] < \infty$

The third condition is known as global identifiability, and 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 model parameters, and that the Jacobian of moments with respect to the model parameters is full-rank.

If in addition the following conditions are true:
(I) (i) The derivative matrix $\nabla_\theta m(x_i,\theta)$ exists and is continuous on Θ for each $x_i \in X$; (ii) θ^* is an interior point of Θ; (iii) $\mathbb{E}_{p(x)}[\nabla_\theta m(x,\theta^*)]$ exists and is finite.

(II) $\mathbb{E}_{p(x)}[m(x,\theta)m(x,\theta)^\intercal]$ exists and is finite, and $\lim_{T\to\infty} cov(T^{1/2}\sum_{i=1}^{N}\frac{m(x_i,\theta^*)}{N}) = \Sigma$ exists and is a finite valued positive definite matrix.

(III) $\mathbb{E}_{p(x)}[\nabla_\theta m(x,\theta)]$ is continuous on some neighborhood $\mathcal{N}_\epsilon$ of θ^*.

(IV) $\sup_{\theta\in\mathcal{N}_\epsilon} \rho \xrightarrow{p} 0$ as $T\to\infty$.
$\rho = tr(\|\frac{1}{T}\sum_{i=1}^{T}\nabla_\theta m(x_i,\theta) - \mathbb{E}_{p(x)}[\nabla_\theta m(x,\theta)]\|^2)^{1/2}$

Then the estimator is asymptotically normal with variance given in Theorem 2.

B.2. Consistency and Asymptotic Normality Conditions for Simulated Method of Moments

Duffie and Singleton (1993) proved consistency and asymptotic normality for the more general case of Markov generators. In the i.i.d. scenario, some of the conditions are trivial. We modify the conditions for the i.i.d. case, but please refer to the original paper for more general conditions.

Consistency conditions are:

- For each $\theta \in \Theta, \{\|\Phi(g_\theta(z_i))\|_{2+\delta}, i = 1,2,\dots\}$ is bounded for some $\delta > 0$. The family $\{\Phi(g_\theta(z_i))\}$ is Lipschitz, uniformly in probability.
- Σ is nonsingular.
- Define $\mathcal{L}^G(\theta) = \hat{m}_N(x_{1,\dots,N},\Phi,\theta)^\intercal \hat{m}_N(x_{1,\dots,N},\Phi,\theta)$. Then $\mathcal{L}^G(\theta^*) < \mathcal{L}^G(\theta)$ for all $\theta \neq \theta^*$.

Asymptotic normality additionally requires:

- (i) θ^* and estimators $\{\hat{\theta}_N\}$ are interior to Θ. (ii) $\Phi(g_\theta(z_i))$ is continuously differentiable with respect to θ for all i. (iii) $\mathbb{E}_{p(z)}[\nabla_\theta \Phi(g_{\theta^*}(z))]$ exists, is finite, and has full rank.
- The family $\{\nabla_\theta \Phi(g_\theta(z_i)), \theta \in \Theta, i = 1,2,\dots\}$ is Lipschitz, uniformly in probability. For all $\theta \in \Theta, \mathbb{E}_{p(z)}[\|\nabla_\theta \Phi(g_\theta(z))\|] < \infty$, and $\theta \mapsto \mathbb{E}_{p(z)}[\nabla_\theta \Phi(g_\theta(z))]$ is continuous.

If the conditions are true, then the asymptotic variance is the one outlined in Theorem 3.

B.3. Moment Matching with Alternative Distances

Adversarial training seems to be performing moment matching with access to a single moment per generator step. Can we say anything how this changes the asymptotics? Presently, no, but we can say something about the asymptotics of matching a finite number of moments with respect to another metric (in this case $\|l\|_\infty$), instead of squared error:

Theorem 4. *Under the Assumptions below, the estimator $\hat{\theta}_N$ converges in probability to θ^*. Furthermore, we have:*

$$\sqrt{N}(\hat{\theta}_N - \theta^*) \to \arg\min_{\zeta} d(Y + G\zeta)$$

where $Y \sim \mathcal{N}(0,\Sigma)$ and $G \coloneqq \mathbb{E}_{p(z)}[\nabla_\theta \Phi(g_{\theta^*}(z))]$

This result is proved in (Han and De Jong, 2004). Asymptotic normality requires conditions on the distance function $\delta(\cdot)$, conditions on the notion of a localized distance, and moment conditions. The conditions on the distance function are:

- $\delta(\cdot)$ is continuous

	Color MNIST	CIFAR-10	CelebA	ImageNet Daisy
Number of objectives N_o	150	250	250	250
Number of moment training steps N_m	100	100	100	100
Number of generating training steps N_g	2,000	2,000	2,000	2,000
Learning rate α	1E-4	1E-4	1E-4	1E-4
Adam β_1/β_2	0.9/0.999	0.9/0.999	0.9/0.999	0.9/0.999
Activation weights	0.0	1E-4	1E-4	1E-4
Norm penalty parameter λ	0.1	1.0	1.0	1.0
Batch size	1000	200	200	200

Table 9. Hyperparameters for different datasets for all experiments except those comparing to adversarial methods.

	DCGAN	Conv
Number of objectives N_o	700	800
Number of moment training steps N_m	50	50
Number of generating training steps N_g	1,000	1,000
Learning rate α	1E-4	1E-4
Adam β_1/β_2	0.9/0.999	0.9/0.999
Activation weights	1E-3	1E-3
Norm penalty parameter λ	0.1	0.1
Generator batch size	200	200
Moment batch size	50	50

Table 10. Hyperparameters for different architectures for GAN comparison on CIFAR-10.

	CIFAR-10
Noise dimension	128
Projection layer size	4×4×512
Conv. transpose layer 1 output size	8×8×256
Conv. transpose layer 2 output size	16×16×128
Conv. transpose layer 3 output size	32×32×64
Stride-1 Conv. layer output size	32×32×3
Output nonlinearity	tanh
Conv. transpose layer kernel size	4×4
Stride-1 Conv. layer kernel size	3×3
Batch norm	Yes
Number of parameters	3,811,907

Table 12. Generator architecture for GAN comparison on CIFAR-10.

	CelebA
Noise dimension	256
Projection layer size	4×4×1024
Conv. transpose layer 1 output size	8×8×512
Conv. transpose layer 2 output size	16×16×256
Conv. transpose layer 3 output size	32×32×128
Output layer size	64×64×3
Output nonlinearity	tanh
Kernel size	5×5
Batch norm	Yes
Number of parameters	20,615,427

Table 11. Generator architecture for large generator parameter experiment.

- $\delta(x) = 0$ iff $x = 0$
- $\delta(x) = \delta(-x)$
- δ satisfies the triangle inequality up to a finite constant locally (in a neighborhood of 0), i.e., there exists an $\epsilon > 0$ such that if $\|x_1\|_1 < \epsilon$ and $\|x_2\|_1 < \epsilon$ then $\delta(x_1 + x_2) \leq M[\delta(x_1) + \delta(x_2)] \ \ \forall x_1, x_2$, for some $M < \infty$.

The authors define a sequence of *localized distance* functions as

$$d_n(x) = \frac{\delta(n^{-1/2}x)}{\delta(n^{-1/2}1)} \ \ n = 1, 2, \ldots$$

Conditions on the localized distance are:

- There is a real function $\phi(\cdot)$ on $\mathbb{R}^q$ such that $\inf_n d_n(x) \geq \phi(x)$, and $\phi(x) \to \infty$ if $|x| \to \infty$.
- d_n converges uniformly in every compact subset of $\mathbb{R}^q$ to a continuous function d.
- $d(z + Bt)$ achieves its minimum at a unique point of $t \in \mathbb{R}^p$ for each $z \in \mathbb{R}^q$ and for any $q \times p$ matrix B with full column rank.

Conditions on the moments (again removing dependence on Φ) are:

- Θ is a compact set.
- $\hat{m}_N(x_{1,\ldots,N}, \theta) = \frac{1}{N}\sum_{i=1}^{N} m(x_i, \theta)$ converges in probability to a nonrandom function $\mu(\theta)$ uniformly on Θ.
- $\mu(\theta) = 0$ iff $\theta = \theta^*$ where θ^* is an interior point of Θ.
- $\hat{G}_N(\theta) = \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta m(x_i, \theta)$ exists and converges in probability to a nonrandom function $G(\theta)$ uniformly in a neighborhood of θ^* and $G(\theta^*)$ has full column rank.
- There exists $\hat{\theta}$ in between θ and θ^* such that .

$$\hat{m}_N(x_{1,\ldots,N}, \theta) = \hat{m}_N(x_{1,\ldots,N}, \theta^*) + \hat{G}_N(\theta)(\theta - \theta^*)$$

  for θ in a neighborhood of θ^*.
- $\sqrt{N}\hat{m}_N(x_{1,\ldots,N}, \theta) \xrightarrow{d} \mathcal{N}(0, \Sigma)$

C. Proofs

C.1. Proof of Proposition 1

The proof of the following statement is sufficiently simple that there is likely an earlier proof. Unfortunately, we could not find a reference, so we are likely re-proving this statement.

Proposition. *Suppose the kernel function $K(x, y) = K(x - y)$ is real, shift-invariant, Bochner integrable, and without loss of generality K(0)=1. Then:*

$$\begin{aligned}\mathbb{E}_{p(x,x')}[K(x,x')] - 2\mathbb{E}_{p(x,y)}[K(x,y)] + \mathbb{E}_{p(y,y')}[K(y,y')] \\ = \mathbb{E}_{p(w)}[(\mathbb{E}_{p(x)}[\cos(\omega^\mathsf{T} x)] - \mathbb{E}_{p(y)}[\cos(\omega^\mathsf{T} y)])^2] \\ + \mathbb{E}_{p(w)}[(\mathbb{E}_{p(x)}[\sin(\omega^\mathsf{T} x)] - \mathbb{E}_{p(y)}[\sin(\omega^\mathsf{T} y)])^2]\end{aligned}$$

where $p(\omega)$ is a probability measure specified by the kernel.

Proof. From Bochner's Theorem for real kernels (Zhao and Meng, 2015):

$$K(x - y) = \mathbb{E}_{p(\omega)}[K(0)\cos(\omega^\mathsf{T}(x - y))]$$

When $K(0) = 1$, $p(\omega)$ is a probability measure. Without loss of generality let $K(0) = 1$. Since the kernel is integrable we can interchange expectations.

$$\begin{aligned}\mathbb{E}[K(x,y)] =& \mathbb{E}_{p(x,y)}[\mathbb{E}_{p(\omega)}[\cos(\omega^\mathsf{T}(x - y))]] \\ =& \mathbb{E}_{p(\omega)}[\mathbb{E}_{p(x,y)}[\cos(\omega^\mathsf{T}(x - y))]]\end{aligned}$$

Then:

$$\begin{aligned}\mathbb{E}_{p(x,y)}[\cos(\omega^\mathsf{T}(x - y))] &= \mathbb{E}_{p(x,y)}[\cos(\omega^\mathsf{T} x)\cos(\omega^\mathsf{T} y)] \\ &+ \mathbb{E}_{p(x,y)}[\sin(\omega^\mathsf{T} x)\sin(\omega^\mathsf{T} y)] \\ &= \mathbb{E}_{p(x)}[\cos(\omega^\mathsf{T} x)]\mathbb{E}_{(y)}[\cos(\omega^\mathsf{T} y)] \\ &+ \mathbb{E}_{p(x)}[\sin(\omega^\mathsf{T} x)]\mathbb{E}_{p(y)}[\sin(\omega^\mathsf{T} y)]\end{aligned}$$

Addition of $\mathbb{E}_{p(x,x')}[K(x,x')] - 2\mathbb{E}_{p(x,y)}[K(x,y)] + \mathbb{E}_{p(y,y')}[K(y,y')]$ yields the result. □

C.2. Simplification of Coulomb GAN

We offer a simpler interpretation of optimality of the generator in Couloumb GAN (Unterthiner et al., 2017) using ideas from Maximum Mean Discrepancy. Suppose we are learning an implicit generative model using MMD:

$$\mathcal{L}(\theta) = \min_\theta \sup_{f \in \mathcal{F}} \mathbb{E}_{p(x)}[f(x)] - \mathbb{E}_{p(z)}[f(g_\theta(z))]$$

If we knew f^*, the function that maximizes the inner supremum, then we can simplify the loss to:

$$\mathcal{L}(\theta) = \min_\theta -\mathbb{E}_{p(z)}[f^*(g_\theta(z))] \tag{3}$$

If the function class $\mathcal{F}$ is the unit ball in a Reproducing Kernel Hilbert Space, then the witness function f^*, defined in Gretton et al. (2012), can be analytically calculated as:

$$f^*(t) \propto \mathbb{E}_{p(x)}[k(x,t)] - \mathbb{E}_{p(z)}[k(g_\theta(z),t)]$$

The empirical version of which is:

$$\hat{f}^*(t) \propto \frac{1}{m}\sum_i k(x_i,t) - \frac{1}{n}\sum_j k(g_\theta(z_j),t)$$

Plugging in this scaled witness function into the Monte Carlo estimate of Equation 3 gives us a biased estimate of the loss. $\mathcal{L}(\theta)$ is a distance if the kernel $k(x, y)$ is characteristic.

In Coulomb GAN (Unterthiner et al., 2017), the discriminator and generator steps are:

$$\begin{aligned}\mathcal{L}_D(D;G) &= \frac{1}{2}\mathbb{E}_{p(t)}\left((D(t) - \hat{\Phi}(t))^2\right) \\ \mathcal{L}_G(D;G) &= -\frac{1}{2}\mathbb{E}_{p(z)}(D(g_\theta(z)))\end{aligned}$$

The authors define the empirical estimate of the potential function Φ (not to be confused with feature functions in the main text) as:

$$\hat{\Phi}(t) = \frac{1}{m}\sum_i k(x_i,t) - \frac{1}{n}\sum_j k(g_\theta(z_j),t)$$

and

$$\begin{aligned}p(t) = &\frac{1}{2}\int \mathcal{N}(t; g_\theta(z), \epsilon I)p_z(z)dz \\ &+ \frac{1}{2}\int \mathcal{N}(t; x, \epsilon I)p_x(x)dx\end{aligned}$$

$\hat{\Phi}$ is merely the empirical estimate of the witness function, discriminator D is a model of the empirical witness function,

and the generator loss is that of Equation 3. The empirical estimate of $\mathcal{L}_G(D; G)$ is biased, though it's unknown how this affects training in practice. Note that dependence of θ on f^* requires frequent retraining of D.

To demonstrate that the loss is a distance, it remains to show that the function class $\mathcal{F}$ is rich enough, or equivalently that the kernel function $k(x, y)$ is characteristic. Note that the proposed Plummer kernel:

$$k_p(a, b) = \frac{1}{(\sqrt{\|a - b\|^2 + \epsilon^2})^d}$$

is a rational quadratic kernel:

$$k_{rq}(a, b) = \sigma^2 \left(1 + \frac{\|a - b\|^2}{2\alpha l^2}\right)^{-\alpha}$$

with $\alpha = \frac{d}{2}$, $\sigma = \epsilon^{-d/2}$ and $l = \frac{\epsilon}{\sqrt{d}}$. Since rational quadratic kernels are characteristic, so are Plummer kernels.