LSGANs with Gradient Regularizers are Smooth High-dimensional Interpolators

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider the problem of discriminator optimization in least-squares generative adversarial networks (LSGANs) subject to higher-order gradient regularization enforced on the convex hull of all possible interpolation points between the target (real) and generated (fake) data. We analyze the proposed LSGAN cost within a *variational framework*, and show that the optimal discriminator solves a regularized least-squares problem, and can be represented through a polyharmonic radial basis function (RBF) interpolator. The optimal RBF discriminator can be implemented in closed-form, with the weights computed by solving a linear system of equations. We validate the proposed approach on synthetic Gaussian and standard image datasets. While the optimal LSGAN discriminator leads to a superior convergence on Gaussian data, the inherent *low-dimensional manifold* structure of images makes the implementation of the optimal discriminator ill-posed. Nevertheless, replacing the trainable discriminator network with a closed-form RBF interpolator results in superior convergence on 2-D Gaussian data, while overcoming pitfalls in GAN training, namely *mode dropping* and *mode collapse*.

1 Introduction

Generative adversarial networks (Goodfellow et al., 2014) are a two-player game involving a generator network G and a discriminator network D. The generator learns to transform samples from a noise distribution, typically Gaussian, into images that follow a distribution p_g. On the other hand, the discriminator accepts an input, either a *fake* sample from p_g, or a *real* sample (image from a dataset) with underlying distribution p_d, and learns to differentiate between the two.

The GAN Discriminator: Broadly speaking, the discriminator has two interpretations. In the standard GAN (SGAN) (Goodfellow et al., 2014), least-squares GAN (LSGAN) (Mao et al., 2017) or the f-GAN (Nowozin et al., 2016), the discriminator has been shown to mimic a chosen divergence metric between the target and generator distributions (*e.g.,* the Jensen-Shannon or the Pearson-χ^2 divergence). The divergence-based approaches fail if the data and generator distributions are non-overlapping (Arjovsky et al., 2017). As an alternative, *integral probability metrics* (IPMs) were proposed, where the discriminator (often called the critic) approximates a distance measure between the distributions, which induces a constraint on the class of critics. For example, the Wasserstein GAN (WGAN) (Arjovsky et al., 2017), which minimizes the *earth mover's* distance, requires a Lipschitz-1 critic. Gulrajani et al. (2017); Bellemare et al. (2017); Mescheder et al. (2018); Mroueh et al. (2018) and Adler & Lunz (2018) consider regularizers on the energy of the gradients of the discriminator. Along another vertical, kernel-based discriminators with Gaussian and inverse multi-quadric kernel have been considered by Li et al. (2015, 2017a). While WGAN variants are a stable choice (Kang et al., 2022), recent works by Jolicoeur-Martineau et al. (2021) have been successfull in employing the LSGAN loss in downstream adversarial score-matching applications.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

LSGANs and Gradient Penalties: Interpolation regularizers have been introduced in WGAN with the gradient penalty. Roth et al. (2017); Kodali et al. (2017); Mescheder et al. (2018) observed that minimizing the norm of the discriminator's gradient (on interpolated points) on divergence-minimizing GANs such as LSGAN results in superior performance. However, neither do the links to Lipschitz-1 discriminators hold, nor does the GAN approximate the Pearson χ^2 divergence. **What does optimizing the LSGAN with gradient penalty actually lead to?** This gap in understanding is what we seek to address in this paper.

1.1 The Proposed Approach

In this paper, we establish an explicit link between discriminator learning and high-dimensional interpolation. Intuitively, given a batch of samples in $\mathbb{R}^n$, the LSGAN discriminator can be seen as assigning specific *class-labels* to the real samples and the fake ones. Given an unseen sample $\boldsymbol{x}$, the ideal discriminator output $D(\boldsymbol{x})$ depends on the values assigned to all points in the vicinity of $\boldsymbol{x}$, which is precisely the task of kernel-based interpolation. Optimization of functions with higher-order derivatives having bounded L_2 norm to interpolate a given set of points has a unique solution (Duchon, 1977; Meinguet, 1979). This has led to successful application of higher-order gradient regularization in image processing tasks – image interpolation (Tirosh et al., 2006) and super-resolution (Ren et al., 2013), to name a few. We therefore consider the gradient-regularized LSGAN cost analyzed from a functional optimization standpoint. Our analysis shows that the optimal discriminator in the proposed setting involves polyharmonic radial basis functions (RBFs) for interpolation. We implement the optimal discriminator through the RBF network with predetermined weights and centers. *Essentially, we show that enforcing interpolation-based gradient regularizers on the LSGAN loss on the space of images, results in an optimal discriminator that performs interpolation on the space of class-labels.* The proposed approach, referred to as *Poly-LSGAN*, outperforms baseline LSGANs in terms of training stability and convergence on Gaussian learning tasks. However, in practice, computing the weights of the interpolator is impractical due to singularity issues arising from the *Manifold Hypothesis*, *i.e.,* images lie in low-dimension embeddings in high-dimensional spaces (Kelley, 2017). Nevertheless, the superior performance on synthetic data makes Poly-LSGAN a promising direction for further research.

2 LSGANs and Gradient Penalties

Mao et al. (2017) considered the GAN learning problem where the discriminator and generator networks minimize the least-squares loss. To mimic the classifier nature of the standard GAN (Goodfellow et al., 2014), an $a-b$ coding scheme is used, where a and b are the class labels of the generated samples and target data samples, respectively. On the other hand, the generator is trained to generate samples that are assigned a class label c by the discriminator. The resulting formulation is as follows:

$$\mathcal{L}_D^{\mathrm{LS}} = \frac{1}{2}\,\mathbb{E}_{\boldsymbol{x}\sim p_d}[(D(\boldsymbol{x})-b)^2] + \frac{1}{2}\,\mathbb{E}_{\boldsymbol{x}\sim p_g}[(D(\boldsymbol{x})-a)^2]\,; \qquad D^*(\boldsymbol{x}) = \arg\min_D \mathcal{L}_D^{\mathrm{LS}},$$

$$\text{and } \mathcal{L}_G^{\mathrm{LS}} = \frac{1}{2}\,\mathbb{E}_{\boldsymbol{x}\sim p_g}[(D^*(\boldsymbol{x})-c)^2]\,; \qquad p_g^*(\boldsymbol{x}) = \arg\min_{p_g} \mathcal{L}_G^{\mathrm{LS}}.$$

While Mao et al. (2017) show that setting $b-a=2$ and $c-a=1$ lead to the generator minimizing the Pearson-χ^2 divergence, a more intuitive approach is to set $c=b$, which enforces the generator to output samples that are classified as *real* by the discriminator. Rosca et al. (2020) showed that the gradient penalties in GANs have the general form $\mathbb{E}_{\boldsymbol{x}\sim p_r}\left[\left(\|\nabla D\left(\boldsymbol{x}\right)\|_2 - K\right)^2\right]$, where p_r is the reference density and K is a suitable constant. Works such as Gulrajani et al. (2017); Petzka et al. (2018); Terjék (2020) consider $K>0$ to enforce Lipschitz smoothness and use $p_r = p_{\mathrm{int}}$, p_{int} being the interpolated distribution $\alpha p_d + (1-\alpha)p_g;\ \alpha\in[0,1]$. On the other hand, Kodali et al. (2017); Mescheder et al. (2018); Mroueh et al. (2018) consider p_r as limiting cases of either p_d or p_g, with $K=0$ to promote the smoothness of the learnt discriminator. Adler & Lunz (2018) implemented m^{th}-order generalizations of the cost empirically through a Fourier representation of the cost.

2.1 Regularized LSGAN and Least-squares Interpolation

The ideal discriminator, upon convergence, will take a constant value for all values of $\boldsymbol{x}$ over $\mathcal{X}$, the convex hull of the supports of p_d and p_g. We set p_r to be the uniform density over $\mathcal{X}$, which implies

weighting all samples drawn from both p_d and p_g, and their possible linear combinations equally. In order to accelerate the convergence of the discriminator during training, we propose to employ higher-order gradient regularization with $K = 0$:

$$\mathcal{L}_D = \mathcal{L}_D^{\text{LS}} + \lambda_D \int_{\mathcal{X}} \|\nabla^m D(\boldsymbol{x})\|_2^2 \, \mathrm{d}\boldsymbol{x}, \tag{1}$$

where $\lambda_d \geq 0$ is the Lagrange multiplier associated with the gradient penalty.

Consider an N-sample approximation of $\mathcal{L}_D^{\text{LS}}$ in Equation (1), where N_B samples are drawn from p_d and p_g each (therefore, $N = 2N_B$), represented by the dataset batch

$$\mathcal{D} = \left\{(\boldsymbol{c}_i, y_i)\right\}_{i=1}^{N} = \left\{(\boldsymbol{x}_i, b) \mid \boldsymbol{x}_i \sim p_d\right\}_{i=1}^{N_b} \bigcup \left\{(\boldsymbol{x}_j, a) \mid \boldsymbol{x}_j \sim p_g\right\}_{j=1}^{N_b}.$$

The corresponding discriminator optimization problem can be formulated as follows:

$$D^* = \arg\min_{D} \sum_{\substack{i=1 \\ (\boldsymbol{c}_i, y_i) \sim \mathcal{D}}}^{N} \left(D(\boldsymbol{c}_i) - y_i\right)^2 + \lambda_D \int_{\mathcal{X}} \|\nabla^m D(\boldsymbol{x})\|_2^2 \, \mathrm{d}\boldsymbol{x}. \tag{2}$$

The above represents a regularized least-squares interpolation problem. When $\lambda_D = 0$, the optimum D^* is an interpolator that passes through the target points $(\boldsymbol{x}_i, y_i)$ exactly. On the other hand, for positive values of λ_D, the minimization leads to smoother solutions, penalizing sharp transitions in the discriminator. We found out experimentally that $\lambda_D = 10$ results in superior performance. A smoother discriminator allows for more efficient training of the generator (Li et al., 2017b; Xu et al., 2018). The following theorem shows that the optimal discriminator that solves the above least-squares minimization problem is an interpolator between the real and fake class labels.

Theorem 2.1. *The **optimal LSGAN discriminator** that minimizes the cost given in Eqn.* (2) *is*

$$D^*(\boldsymbol{x}) = \sum_{\substack{1=i \\ (\boldsymbol{c}_i, y_i) \sim \mathcal{D}}}^{N} w_i \psi_k\left(\|\boldsymbol{x} - \boldsymbol{c}_i\|\right) + \mathcal{P}_{m-1}(\boldsymbol{x}; \boldsymbol{v}), \tag{3}$$

$$\textit{where} \quad \psi_k(r) = \begin{cases} r^k & \textit{for } \ k = 1, 3, 5, \cdots \\ r^k \ln(r) & \textit{for } \ k = 2, 4, 6, \cdots \end{cases} \tag{4}$$

is the polyharmonic radial basis function, $\mathcal{P}_{m-1}(\boldsymbol{x}; \boldsymbol{v})$ is an $(m-1)^{th}$ order polynomial parametrized by the coefficients $\boldsymbol{v} \in \mathbb{R}^L$, $\boldsymbol{x} \in \mathbb{R}^n$, $\mathcal{D} = \{(\boldsymbol{c}_i, y_i)\}$ is the set of real and fake centers about which the polyharmonic RBFs $\psi_k(\|\cdot\|)$ are localized, $\|\cdot\|$ denotes the ℓ_2 norm, and k denotes the order of the polyharmonic interpolator, which implicitly assumes a gradient penalty of order $m = \lceil \frac{k+n}{2} \rceil$. The N weights $\boldsymbol{w} = [w_1, w_2, \ldots, w_N]^{\mathrm{T}}$ and L polynomial coefficients $\boldsymbol{v} = [v_1, v_2, \ldots, v_L]^{\mathrm{T}}$ can be obtained by solving the linear system of equations:

$$\begin{bmatrix} \mathbf{A} + (-1)^m \lambda_D \mathrm{C}_k \mathbf{I} & \mathbf{B} \\ \mathbf{B}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} \\ \mathbf{0} \end{bmatrix} \quad , \tag{5}$$

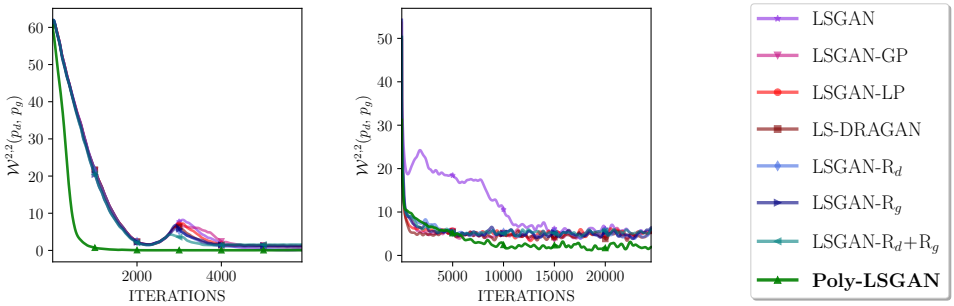
$$\textit{where } [\mathbf{A}]_{i,j} = \psi_k(\|\boldsymbol{c}_i - \boldsymbol{c}_j\|), \ \mathbf{B} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_N \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{c}_1^{m-1} & \boldsymbol{c}_2^{m-1} & \cdots & \boldsymbol{c}_N^{m-1} \end{bmatrix}^{\mathrm{T}}, \textit{and } \ \boldsymbol{y} = [y_1, y_2, \cdots, y_N]^{\mathrm{T}},$$

$\mathbf{I}$ is the $N \times N$ identity matrix, and $\boldsymbol{c}_i^j$ is a vector of all j^{th} order monomials of $\boldsymbol{c}_i$. The matrix B is required to be full-rank for invertibility, and C_k is a constant that depends only on the order k.

The proof follows by applying the Euler-Lagrange equation from the *Calculus of Variations* to the cost in Equation (2). The details are provided in Appendix B. Owing to the polyharmonic radial basis kernel, the proposed approach is referred to as *Poly-LSGAN*.

3 Experimental Evaluation

We now evaluate the optimal Poly-LSGAN discriminator on learning synthetic 2-D Gaussian and Gaussian mixture models (GMMs), and subsequently discuss extensions to learning images. We consider polyharmonic spline order $k = 1, 2$. For larger k, we encountered numerical instability.



(a) (b)

Figure 1: The Wasserstein-2 distance versus iterations on learning (a) a 2-D Gaussian; and (b) a 2-D Gaussian mixture, for various LSGAN variants. The performance of the Poly-LSGAN with the RBF discriminator is superior to the baselines in both scenarios. The convergence is also relatively smoother and stabler, unlike the baselines, which have fluctuations on the 2-D Gaussian learning task.

3.1 Validation on Synthetic Data

We compare Poly-LSGAN against the base LSGAN (Mao et al., 2017), and LSGAN subjected to the gradient penalty (GP) (Gulrajani et al., 2017) , R$_d$ and R$_g$ (Mescheder et al., 2018), Lipschitz penalty (LP) (Petzka et al., 2018) and the DRAGAN (Roth et al., 2017) regularizers. On the unimodal learning task, the target is $\mathcal{N}(5\mathbf{1}_2, 1.5\mathbb{I}_2)$, where $\mathbf{1}_2$ is the 2-D vector of ones, and $\mathbb{I}_2$ is the 2×2 identity matrix. On the GMM learning task, we consider eight components distributed uniformly about the unit circle, each having a standard deviation of 0.02. To quantify performance, we use the Wasserstein-2 distance between the target and generator distributions $\left(\mathcal{W}^{2,2}(p_d, p_g)\right)$. Network parameters are given in Appendix C. Figure 1 presents the $\mathcal{W}^{2,2}$ distance as a function of iterations on the Gaussian and Gaussian mixture learning tasks. On both datasets, we observe that using polyharmonic RBF discriminator results in superior generator performance (lower $\mathcal{W}^{2,2}$ scores). In all scenarios considered, the polyharmonic RBF discriminator learns the perfect classifier, compared to LSGAN with a trainable network discriminator. Additional comparisons are given in Appendix D.

3.2 Experiments on Image Datasets

We train on MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017) and CelebA (Liu et al., 2015) datasets. The results are presented in Appendix D. While the underlying structure is learnt, the generated images are far from being realistic and below par compared with standard GAN results. Poly-LSGAN failed to converge as the monomial matrix $\mathbf{B}$ (which can be viewed as a vector generalization of the Vandermonde matrix) became singular. As noted in the literature on mesh-free interpolation (Iske, 2004), $\mathbf{B}$ must be full-rank, for the system of equations (Eq. (5)) to have a unique solution. This requires the centers $\boldsymbol{c}_i$ to not lie on a subspace/manifold of $\mathbb{R}^n$. However, from the manifold hypothesis (Kelley, 2017; Vershynin, 2018), we know that structured image datasets lie precisely in such low-dimensional manifolds. To ascertain this behavior, we simulated a data manifold with $n = 32 \times 32 \times 3 = 3072$ dimensions, with the first M entries distributed as $\mathcal{N}(0.5\mathbf{1}_M, 0.2\mathbb{I})$ and the rest zeros, and observed that for $M \neq n$, the matrix $\mathbf{B}$ indeed became singular.

4 Conclusions

Considering a generalization of the gradient-regularized LSGAN cost, we showed that the optimal discriminator solves the regularized least-squares interpolation problem, whose solution can be represented using the polyharmonic RBF. The proposed discriminator results in superior performance on synthetic datasets such as 2-D Gaussians and Gaussian mixture densities. However, the rank deficiency of the monomial matrix $\mathbf{B}$ caused convergence issues on image datasets. Although implementing the optimal discriminator yields superior performance in idealistic settings, a straightforward extension to image generation in the ambient dimension became intractable due to singularity issues. One approach to overcoming this issue is to resort to a latent space solution as in the case of Wasserstein autoencoders (Tolstikhin et al., 2018), or to perform adversarial score matching (Jolicoeur-Martineau et al., 2021). Alternatively, one could also analyze the effect of enforcing the gradient-based regularizers on other GAN losses, such as the vanilla GAN or WGAN.

References

Abadi, M. et al. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint, arXiv:1603.04467*, Mar. 2016. URL `https://arxiv.org/abs/1603.04467`.

Adler, J. and Lunz, S. Banach Wasserstein GAN. In *Advances in Neural Information Processing Systems 31*, pp. 6754–6763. 2018.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 214–223, 2017.

Aronszajn, N., Creese, T., and Lipkin, L. *Polyharmonic Functions*. Oxford: Clarendon, 1983.

Bellemare, M. G., Danihelka, I., Dabney, W., Mohamed, S., Lakshminarayanan, B., Hoyer, S., and Munos, R. The Cramér distance as a solution to biased Wasserstein gradients. 2017.

Duchon, J. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive Theory of Functions of Several Variables*, pp. 85–100. Springer, 1977.

Fasshauer, G. E. *Meshfree Approximation Methods with MATLAB*. World Scientific, 2007.

Flamary, R. et al. POT: Python optimal transport. *Journal of Machine Learning Research*, 22(78): 1–8, 2021. URL `http://jmlr.org/papers/v22/20-451.html`.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. 2014.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems 30*, pp. 5767–5777. 2017.

Iske, A. Multiresolution methods in scattered data modelling. In *Lecture Notes in Computational Science and Engineering*, 2004.

Jolicoeur-Martineau, A., Piché-Taillefer, R., Mitliagkas, I., and des Combes, R. T. Adversarial score matching and improved sampling for image generation. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=eLfqMl3z3lq`.

Kang, M., Shin, J., and Park, J. StudioGAN: A taxonomy and benchmark of GANs for image synthesis, 2022. URL `https://arxiv.org/abs/2206.09479`.

Kelley, J. L. *General Topology*. Courier Dover Publications, Inc., 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

Kodali, N., Abernethy, J. D., Hays, J., and Kira, Z. On convergence and stability of GANs. *arXiv preprint, arXiv:1705.07215*, May 2017. URL `http://arxiv.org/abs/1705.07215`.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, C. L., Chang, W. C., Cheng, Y., Yang, Y., and Poczos, B. MMD GAN: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems 30*, pp. 2203–2213. 2017a.

Li, J., Madry, A., Peebles, J., and Schmidt, L. Towards understanding the dynamics of generative adversarial networks. *arXiv preprints, arXiv:1706.09884*, 2017b. URL `http://arxiv.org/abs/1706.09884`.

Li, Y., Swersky, K., and Zemel, R. Generative moment matching networks. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1718–1727, Jul 2015.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision*, 2015.

Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., and Smolley, S. P. Least squares generative adversarial networks. In *Proceedings of International Conference on Computer Vision*, 2017.

Meinguet, J. Multivariate interpolation at arbitrary points made simple. *Journal of Applied Mathematics and Physics (ZAMP)*, 30(2):292–304, 1979.

Mescheder, L., Geiger, A., and Nowozin, S. Which training methods for GANs do actually converge? In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 3481–3490, 2018.

Mroueh, Y., Li, C., Sercu, T., Raj, A., and Cheng, Y. Sobolev GAN. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

Nowozin, S., Cseke, B., and Tomioka, R. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems 29*, pp. 271–279. 2016.

Petzka, H., Fischer, A., and Lukovnikov, D. On the regularization of Wasserstein GANs. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proceedings of the 4th International Conference on Learning Representations*, 2016.

Ren, Z., He, C., and Zhang, Q. Fractional-order total-variation regularization for image super-resolution. *Signal Processing*, 93(9):2408–2421, 2013.

Rosca, M., Weber, T., Gretton, A., and Mohamed, S. A case for new neural network smoothness constraints. In *Proceedings on "I Can't Believe It's Not Better!" at NeurIPS Workshops*, Proceedings of Machine Learning Research, 12 Dec 2020.

Roth, K., Lucchi, A., Nowozin, S., and Hofmann, T. Stabilizing training of generative adversarial networks through regularization. In *Advances in Neural Information Processing Systems 30*, pp. 2015–2025, 2017.

Terjék, D. Adversarial Lipschitz regularization. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

Tirosh, S., De Ville, D. V., and Unser, M. Polyharmonic smoothing splines and the multidimensional Wiener filtering of fractal-like signals. *IEEE Transactions on Image Processing*, 15(9):2616–2630, 2006.

Tolstikhin, I. O., Bousquet, O., Gelly, S., and Schölkopf, B. Wasserstein auto-encoders. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

Vershynin, R. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint, arXiv:1708.07747*, Aug. 2017. URL `https://arxiv.org/abs/1708.07747`.

Xu, Z., Li, C., and Jegelka, S. Robust GANs against dishonest adversaries, 2018. URL `https://arxiv.org/abs/1802.09700`.

A Mathematical Preliminaries

We recall the high-dimensional Euler-Lagrange conditions from the Calculus of Variations, which play an important role in the optimization of the Poly-LSGAN cost in Equation 2.

Consider a vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^{\mathrm{T}} \in \mathbb{R}^n$ and a function $f : \mathbb{R}^n \to \mathbb{R}$. The notation $\nabla^m f(\boldsymbol{x})$ denotes the vector of mth-order partial derivatives of f with respect to the entries of $\boldsymbol{x}$. ∇^0 is the identity operator. The elements of $\nabla^m f$ are represented using the multi-index notation $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n]^{\mathrm{T}}$, as:

$$\partial^{\boldsymbol{\alpha}} f = \frac{\partial^{|\boldsymbol{\alpha}|}}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \ldots \partial x_n^{\alpha_n}} f, \text{ where } |\boldsymbol{\alpha}| = \sum_{i=1}^{n} \alpha_i.$$

and $\boldsymbol{\alpha} \in \mathbb{Z}_*^n$, the set of n-dimensional vectors with non-negative integer entries. For example, with $n = 4, m = 3$, the index $\boldsymbol{\alpha} = [2, 0, 1, 0]^{\mathrm{T}}$ yields the element $\frac{\partial^3}{\partial x_1^2 \partial x_3} f(\boldsymbol{x})$. The square of the L_2-norm of $\nabla^m f$ is given by a multidimensional sum:

$$\|\nabla^m f(\boldsymbol{x})\|_2^2 = \sum_{\boldsymbol{\alpha}:\, |\boldsymbol{\alpha}|=m} \left(\frac{m!}{\boldsymbol{\alpha}!}\right) \left(\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x})\right)^2, \tag{6}$$

where $\boldsymbol{\alpha}! = \alpha_1! \alpha_2! \ldots \alpha_n!$. The iterated Laplacian, also known as the polyharmonic operator, is defined as: $\Delta^m f(\boldsymbol{x}) = \Delta(\Delta^{m-1} f(\boldsymbol{x}))$, where $\Delta f(\boldsymbol{x}) = \nabla . \nabla f(\boldsymbol{x}) = \sum_{i=1}^{n} \frac{\partial^2}{\partial x_i^2} f(\boldsymbol{x})$ is the Laplacian operator acting on $f(\boldsymbol{x})$. Applying the multi-index notation yields the standard form of the polyharmonic operator:

$$\Delta^m f(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha}:\, |\boldsymbol{\alpha}|=m} \left(\frac{m!}{\boldsymbol{\alpha}!}\right) \partial^{\boldsymbol{\alpha}} \left(\partial^{\boldsymbol{\alpha}} f(\boldsymbol{x})\right).$$

Consider an integral cost $\mathcal{L}$ with the integrand $\mathcal{F}$ dependent on f and all its partial derivatives up to and including order ℓ, given by $\mathcal{L}\left(f(\boldsymbol{x}), \partial^{\boldsymbol{\alpha}} f; |\boldsymbol{\alpha}| \leq \ell\right) = \int_{\mathcal{X}} \mathcal{F}\left(f(\boldsymbol{x}), \partial^{\boldsymbol{\alpha}} f; |\boldsymbol{\alpha}| \leq \ell\right) \mathrm{d}\boldsymbol{x}$, defined on a suitable domain $\mathcal{X}$ over which f and its partial derivatives up to and including order ℓ are continuously differentiable. The optimizer f^* must satisfy the Euler-Lagrange condition:

$$\frac{\partial \mathcal{F}}{\partial f} + \sum_{j=1}^{\ell} \left((-1)^j \sum_{\boldsymbol{\alpha}:\, |\boldsymbol{\alpha}|=j} \partial^{\boldsymbol{\alpha}} \left(\frac{\partial \mathcal{F}}{\partial (\partial^{\boldsymbol{\alpha}} f)} \right) \right) \Bigg|_{f=f^*} = 0 \tag{7}$$

B The Optimal Poly-LSGAN Discriminator

The proof of Theorem 2.1 follows from the results in mesh-free interpolation literature (Aronszajn et al., 1983; Iske, 2004; Fasshauer, 2007) that deal with the generic polyharmonic spline interpolation problem. For completeness, we provide the proof here. While the assumption may appear strong, we show that this is implicitly satisfied by the optimal solution. Recall the discriminator optimization problem given in Eq. (2)

$$\arg\min_{D} \left\{ \sum_{\substack{i=1 \\ (\boldsymbol{c}_i, y_i) \sim \mathcal{D}}}^{N} \left(D(\boldsymbol{c}_i) - y_i\right)^2 + \lambda_D \int_{\mathcal{X}} \|\nabla^m D(\boldsymbol{x})\|_2^2 \, \mathrm{d}\boldsymbol{x} \right\}. \tag{8}$$

To compute the functional optimum in the Calculus of Variations setting, the above cost must be cast into an integral form. Using the Dirac delta function, we have:

$$\sum_{\substack{i=1 \\ (\boldsymbol{c}_i, y_i) \sim \mathcal{D}}}^{N} \left(D(\boldsymbol{c}_i) - y_i\right)^2 = \int_{\mathcal{X}} \sum_{\substack{i=1 \\ (\boldsymbol{c}_i, y_i) \sim \mathcal{D}}}^{N} \left(D(\boldsymbol{x}) - y_i\right)^2 \delta(\boldsymbol{x} - \boldsymbol{c}_i) \, \mathrm{d}\boldsymbol{x}.$$

Then, Equation 8 can be rewritten as an integral-cost minimization:

$$\arg\min_{D}\left\{\int_{\mathcal{X}}\underbrace{\sum_{\substack{i=1\\(\boldsymbol{c}_i,y_i)\sim\mathcal{D}}}^{N}\left(D(\boldsymbol{x})-y_i\right)^2\delta(\boldsymbol{x}-\boldsymbol{c}_i)+\lambda_D\|\nabla^m D(\boldsymbol{x})\|_2^2}_{\mathcal{F}(D,\partial^{\boldsymbol{\alpha}}D;\,|\boldsymbol{\alpha}|=m)}\,\mathrm{d}\boldsymbol{x}\right\}.$$

Computing the derivatives of the integrand $\mathcal{F}$ with respect to D and $\partial^\alpha D$ yields

$$\frac{\partial\mathcal{F}}{\partial D}=2\sum_{\substack{i=1\\(\boldsymbol{c}_i,y_i)\sim\mathcal{D}}}^{N}\left(D(\boldsymbol{x})-y_i\right)\delta(\boldsymbol{x}-\boldsymbol{c}_i),\qquad\text{and}\qquad\sum_{\boldsymbol{\alpha}:|\boldsymbol{\alpha}|=m}\partial^{\boldsymbol{\alpha}}\left(\frac{\partial\mathcal{F}}{\partial(\partial^{\boldsymbol{\alpha}}D)}\right)=2\lambda_D\Delta^m D.$$

Substituting the above into the Euler-Lagrange equation (Eq. 7) gives us the partial differential equation that the optimal discriminator $D^*(\boldsymbol{x})$ must satisfy:

$$\left(\sum_{i=1}^{N}(D(\boldsymbol{x})-y_i)\delta(\boldsymbol{x}-\boldsymbol{c}_i)\right)+(-1)^m\lambda_D\Delta D(\boldsymbol{x})\;\Bigg|_{D=D^*(\boldsymbol{x})}=0.$$

While the above condition is applicable for a strong solution, a weak solution to $D(\boldsymbol{x})$ satisfies:

$$\int_{\mathcal{X}}\left(\left(\sum_{i=1}^{N}(D(\boldsymbol{x})-y_i)\delta(\boldsymbol{x}-\boldsymbol{c}_i)\right)+(-1)^m\lambda_D\Delta D(\boldsymbol{x})\right)\eta(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\;\Bigg|_{D=D^*(\boldsymbol{x})}=0,\tag{9}$$

where $\eta(\boldsymbol{x})$ is any test function drawn from the family of compactly-supported infinitely-differentiable functions. Aronszajn et al. (1983), an authoritative resource on polyharmonic functions, has shown that, functions of the form

$$f(\boldsymbol{x})=\sum_{\substack{i=1\\(\boldsymbol{c}_i,y_i)\sim\mathcal{D}}}^{N}w_i\psi_k\left(\|\boldsymbol{x}-\boldsymbol{c}_i\|\right)+\mathcal{P}_{m-1}(\boldsymbol{x};\boldsymbol{v}),\ \text{where}\ \psi_k(r)=\begin{cases}r^k & \text{for}\ \ k=1,3,\cdots\\ r^k\ln(r) & \text{for}\ \ k=2,4,\cdots,\end{cases}\tag{10}$$

satisfy the polyharmonic PDE:

$$\Delta^m f(\boldsymbol{x})=\sum_{i=1}^{N}C_k w_i\delta(\boldsymbol{x}-\boldsymbol{c}_i),$$

where $\mathcal{P}_{m-1}(\boldsymbol{x};\boldsymbol{v})$ is the $(m-1)^{th}$ order polynomial parametrized by the coefficients $\boldsymbol{v}\in\mathbb{R}^L$. For example, with $m=2$, we have $\mathcal{P}(\boldsymbol{x};\boldsymbol{v})=\langle\tilde{\boldsymbol{v}},\boldsymbol{x}\rangle;\ \boldsymbol{v}\in\mathbb{R}^{n+1}$. Substituting the above back into Equation 9, we get

$$\int_{\mathcal{X}}\left(\sum_{i=1}^{N}\left((D(\boldsymbol{x})-y_i)+(-1)^m\lambda_D C_k w_i\right)\delta(\boldsymbol{x}-\boldsymbol{c}_i)\right)\eta(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\;\Bigg|_{D=D^*(\boldsymbol{x})}=0$$

$$\Rightarrow\sum_{i=1}^{N}\left((D(\boldsymbol{c}_i)-y_i)+(-1)^m\lambda_D C_k w_i\right)\eta(\boldsymbol{c}_i)\;\Bigg|_{D=D^*(\boldsymbol{x})}=0$$

Since the above condition must hold for all possible test functions η, we have:

$$D^*(\boldsymbol{c}_i)-y_i+(-1)^m\lambda_D\mathrm{C}_k w_i=0\qquad\forall\,i=1,2,\cdots N,$$

where D^* is given by Equation (10). Substituting for D^* and stacking for all i gives the following condition that the weights and polynomial coefficients satisfy:

$$\left(\mathbf{A}+(-1)^m\lambda_D\mathrm{C}_k\mathbf{I}\right)\boldsymbol{w}+\mathbf{B}\boldsymbol{v}=\boldsymbol{y},\tag{11}$$

$$\text{where}\quad[\mathbf{A}]_{i,j}=\psi_k(\|\boldsymbol{c}_i-\boldsymbol{c}_j\|);\quad\boldsymbol{w}=[w_1,w_2,\ldots,w_N]^{\mathrm{T}},\quad\boldsymbol{y}=[y_1,y_2,\cdots,y_N]^{\mathrm{T}},$$

$$\mathbf{B}=\begin{bmatrix}1&1&\cdots&1\\ \boldsymbol{c}_1&\boldsymbol{c}_2&\cdots&\boldsymbol{c}_N\\ \vdots&\vdots&\ddots&\vdots\\ \boldsymbol{c}_1^{m-1}&\boldsymbol{c}_2^{m-1}&\cdots&\boldsymbol{c}_N^{m-1}\end{bmatrix}^{\mathrm{T}},\ \text{and}\ \boldsymbol{v}=[v_0,v_1,v_2,\ldots,v_L]^{\mathrm{T}}.$$

The matrix $\mathbf{B}$ corresponds to a Vandermonde matrix when $n = 1$. However, the system of linear equations is underdetermined, with fewer equations than unknowns.

To derive the second condition present in Equation (5), $\mathbf{B}^{\mathrm{T}}\boldsymbol{w} = \mathbf{0}$, we enforce the higher-order gradient penalty on the optimal solution D^*. To define the higher-order gradient-norm, we first consider the inner-product space associated with the higher-order gradient, where:

$$\begin{aligned}\langle f, g\rangle &= \int_{\mathbb{R}^n} (\nabla^m f)\cdot(\nabla^m g)\ \mathrm{d}\boldsymbol{x}.\\ &= (-1)^m \int_{\mathbb{R}^n} f\cdot(\Delta^m g)\,\mathrm{d}\boldsymbol{x}\end{aligned}$$

where the second inequality is via integration by parts. For any function D^* of the form given in Equation (10), we have:

$$\begin{aligned}\langle D^*, D^*\rangle &= (-1)^m \int_{\mathbb{R}^n} D^*(\boldsymbol{x})\left(\sum_{i=1}^{N}(-1)^m C_k w_i \delta(\boldsymbol{x}-\boldsymbol{c}_i)\right)\ \mathrm{d}\boldsymbol{x}\\ &= (-1)^m C_k \sum_{i=1}^{N} w_i D^*(\boldsymbol{c}_i).\end{aligned}$$

Substituting for D^* from Equation (10) gives:

$$\begin{aligned}\langle D^*, D^*\rangle = \int_{\mathbb{R}^n} \|\nabla^m D^*\|_2^2\,\mathrm{d}\boldsymbol{x} &= (-1)^m C_k \sum_{i=1}^{N}\left(w_i \sum_{\substack{j=1\\(\boldsymbol{c}_j,y_j)\sim\mathcal{D}}}^{N} w_j \psi_k\left(\|\boldsymbol{c}_j-\boldsymbol{c}_i\|\right) + \mathbf{B}\boldsymbol{v}\right)\\ &= (-1)^m C_k \boldsymbol{w}^{\mathrm{T}}\mathbf{A}\boldsymbol{w},\end{aligned} \tag{12}$$

where the second equality holds as a result of Equation 14. Substituting in D^* and Equation 12 into the optimization problem in Equation 8 yields:

$$\begin{aligned}&\arg\min_{D}\left\{\sum_{\substack{i=1\\(\boldsymbol{c}_i,y_i)\sim\mathcal{D}}}^{N}\left(D(\boldsymbol{c}_i)-y_i\right)^2 + \lambda_D\int_{\mathcal{X}}\|\nabla^m D(\boldsymbol{x})\|_2^2\,\mathrm{d}\boldsymbol{x}\right\}\\ &= \arg\min_{\boldsymbol{w},\boldsymbol{v}}\left\{\underbrace{\|\mathbf{A}\boldsymbol{w}+\mathbf{B}\boldsymbol{v}-\boldsymbol{y}\|_2^2 + \lambda_D C_k \boldsymbol{w}^{\mathrm{T}}\mathbf{A}\boldsymbol{w}}_{\mathrm{F}(\boldsymbol{w},\boldsymbol{v})}\right\}.\end{aligned} \tag{13}$$

Minimizing the cost function in Equation (13) with respect to $\boldsymbol{w}$ and $\boldsymbol{v}$ yields:

$$\begin{aligned}\frac{\partial \mathrm{F}}{\partial \boldsymbol{w}} = 2\mathbf{A}^{\mathrm{T}}\left(\mathbf{A}\boldsymbol{w}+\mathbf{B}\boldsymbol{v}-\boldsymbol{y}+2\lambda_D C_k\boldsymbol{w}\right) = 0, \quad \text{and} \quad \frac{\partial \mathrm{F}}{\partial \boldsymbol{v}} = 2\mathbf{B}^{\mathrm{T}}\left(\mathbf{A}\boldsymbol{w}+\mathbf{B}\boldsymbol{v}-\boldsymbol{y}\right) = 0\\ \Rightarrow \mathbf{B}^{\mathrm{T}}\boldsymbol{w} = \mathbf{0},\end{aligned} \tag{14}$$

which gives us the second necessary condition that the optimal weights and polynomial coefficients must satisfy. Equation (14) ensure that the solution obtained is such that the sum of the unbounded polyharmonic kernels vanish as $\boldsymbol{x}$ tends to infinity. Essentially, in regions close to the centers $\boldsymbol{c}_i$, there is a large contribution in $D^*(\boldsymbol{x})$ from the kernel function, and when far away from the centers, the polynomial has a large contribution in $D^*(\boldsymbol{x})$. This ensures that the the discriminator obtained by solving the system of equations does not grow to infinity. This completes the proof of Theorem 2.1.

C Training Specifications

Experiments on 2-D Gaussian Data: On the unimodal Gaussian learning task, the generator is a single layer affine transformation of the noise $\boldsymbol{z}$, given by $\boldsymbol{x} = M\boldsymbol{z} + b$, while on the GMM task, it is a three-layer neural network with Leaky ReLU activations. The discriminator in baseline LSGANs variants is a three-layer neural network with Leaky ReLU activation in both the Gaussian and GMM learning tasks. Poly-LSGAN employs the RBF discriminator while weights are computed by solving the system of equation given in Equation (5).

The networks are trained using the Adam optimizer (Kingma & Ba, 2015) with a learning rate of $\eta_g = 0.002$ for the generator and $\eta_d = 0.0075$ for the discriminator. A batch size of 500 is employed. The models are evaluated using the Wasserstein-2 distance, $\mathcal{W}^{2,2}(p_d, p_g)$, between the target and generator distributions. The distance $\mathcal{W}^{2,2}(p_d, p_g)$ is computed in closed-form on Gaussian data, given by:

$$\mathcal{W}^{2,2}(p_d, p_g) = \|\boldsymbol{\mu}_d - \boldsymbol{\mu}_g\|^2 + \text{Trace}\left(\Sigma_d + \Sigma_g - 2\sqrt{\Sigma_d \Sigma_g}\right),$$

On the Gaussian mixture data, $\mathcal{W}^{2,2}(p_d, p_g)$ is estimated empirically using the *Python optimal transport* (Flamary et al., 2021) library.

Experiments on Image Data: On image learning tasks, we employ the DCGAN (Radford et al., 2016) generator, trained using the Adam optimizer. The batch size is set to 100. The generator learning rate is set to $\eta_g = 10^{-4}$. The discriminator is the polyharmonic RBF with $k = 1$, which corresponds to enforcing a gradient penalty of order $m = \lceil \frac{n+1}{2} \rceil$. Consequently, there matrix $\mathbf{B}$ contains monomials up to oder $m - 1$.

Codebase: To preserve the anonymity of the *double-blind* review process, the TensorFlow 2.0 (Abadi et al., 2016) based implementation for Poly-LSGAN will be made available publicly on GitHub, subsequent to the review period of this manuscript.

D Additional Experimental Results

We now present additional experiments results from the authentic Gaussian and image datasets experiments conducted on Poly-LSGAN.

Experiments on Gaussian Data: Figures 2 and 3 present the generated and target data samples, superimposed on the level-sets of the discriminator, for the 2-D Gaussian, and 8-component Gaussian mixture learning tasks, respectively. For the Gaussian learning problem, we observe that Poly-LSGAN does not *mode collapse* upon convergence to the target distribution. However, in the baseline GANs, depending on the learning rate, the generator converges to a distribution of smaller support than the target, before latching on to the desired target. Similarly, on the GMM learning task, Poly-LSGAN learns the target distribution more accurately compared to the baselines.

Experiments on Image Data: Figure 4 depicts the images generated by Poly-LSGAN when trained on the MNIST, Fashion-MNIST and CelebA datasets. In all scenarios, we observe that, although the generator is able to generate images from the target dataset, the visual quality of the images is sub-par compared to standard GAN approaches. Additional training of these models resulted in gradient explosion caused by the singularity of the monomial matrix $\mathbf{B}$.

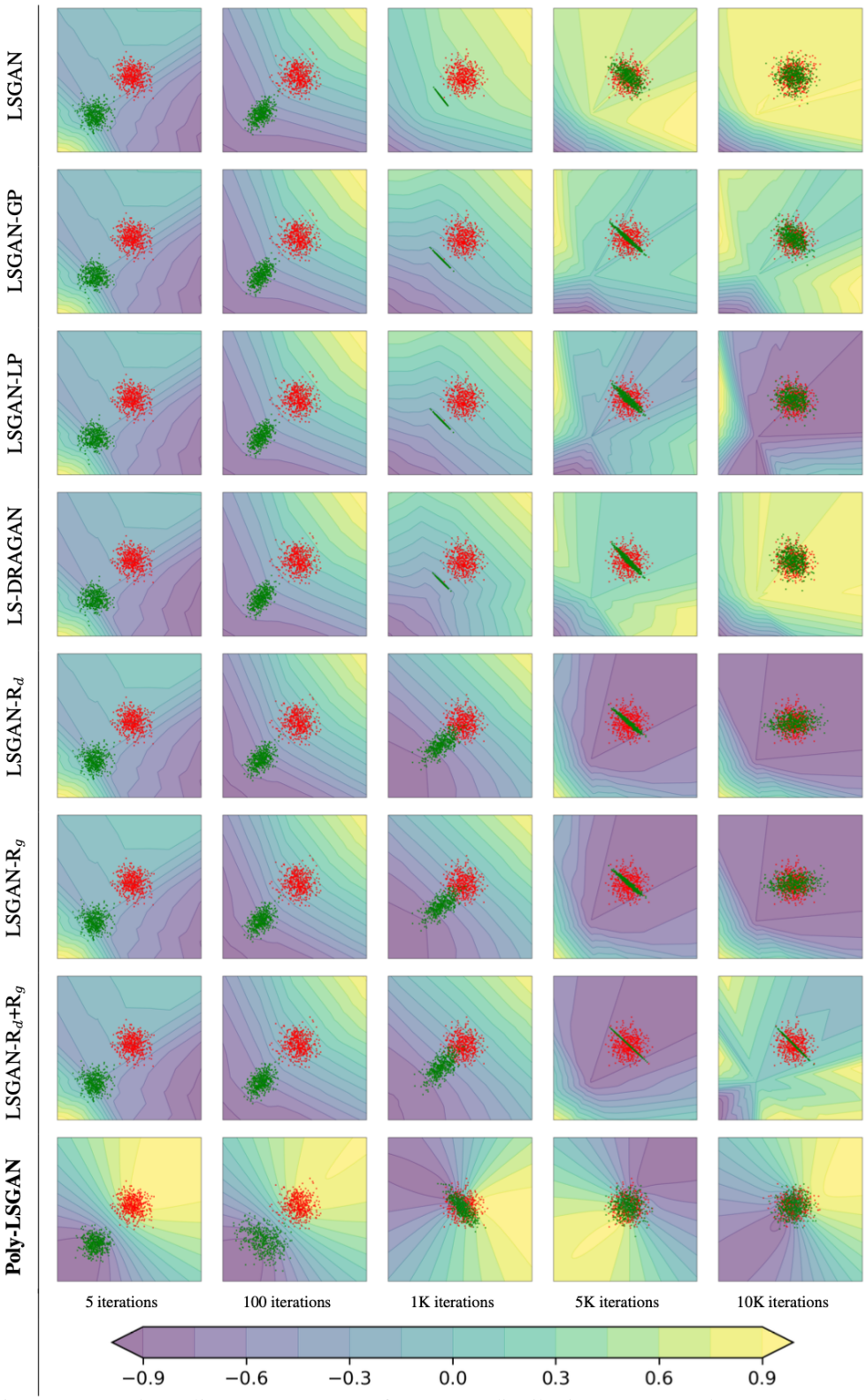


Figure 2: (Color online) Convergence of generator distribution (*green*) to the target 2-D Gaussian data (*red*) on the considered LSGAN variants. The heatmap represents the values taken by the discriminator. The Poly-LSGAN approach leads to a better representation of the discriminator function during the initial training iterations when compared to baseline approaches, leading to a faster convergence. Poly-LSGAN also does not experience *mode collapse*.

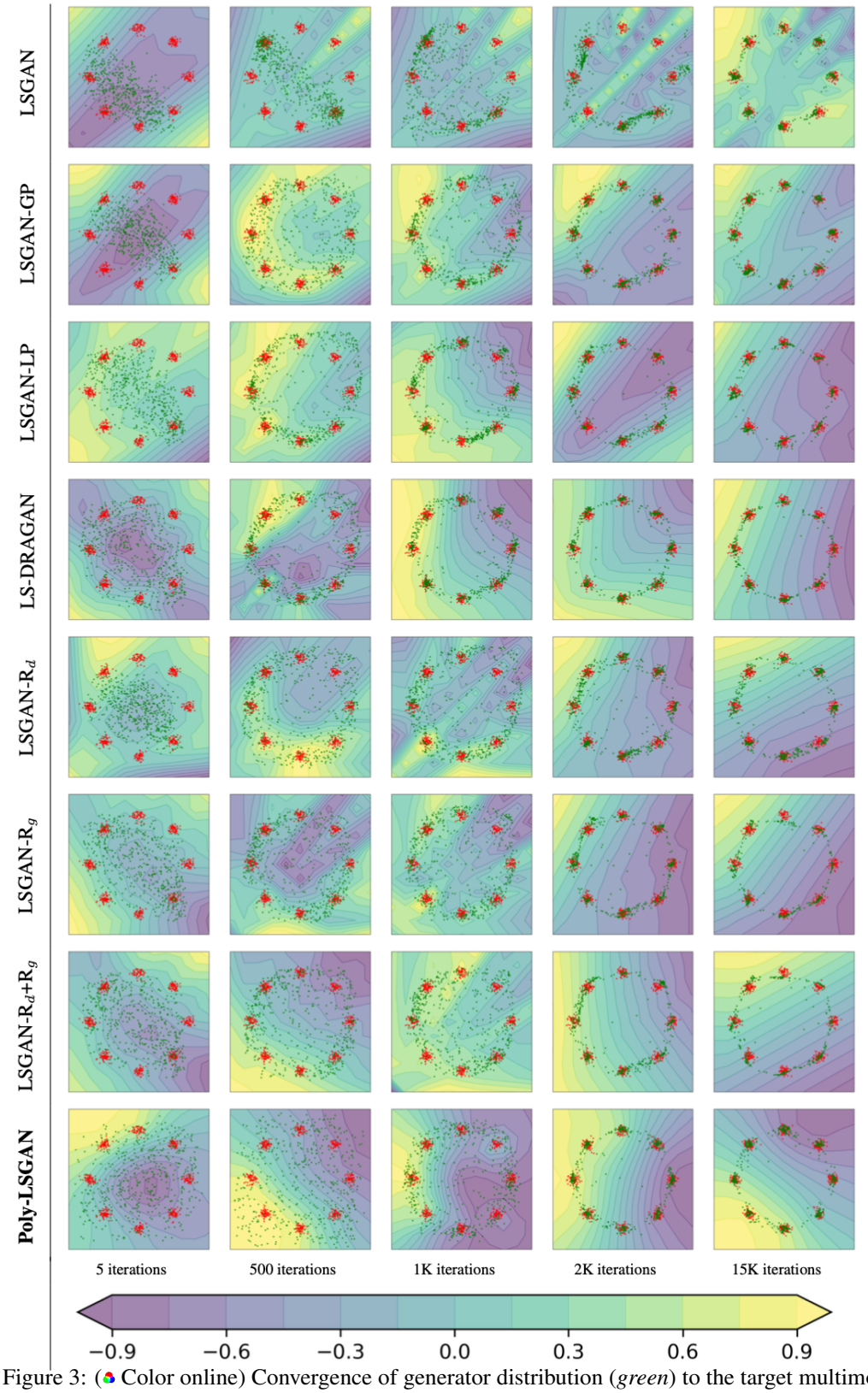


Figure 3: (🎨 Color online) Convergence of generator distribution (*green*) to the target multimodal Gaussian data (*red*) on the considered LSGAN variants, superimposed on the level-sets of the discriminator. The ideal $D(\boldsymbol{x})$ assigned a value of $b = 1$ to reals and $a = -1$ to fakes. Poly-LSGAN is able to identify the modes of the GMM more accurately than the baselines.

Figure 4: Images generated by training Poly-LSGAN on vectorized images drawn from (a) MNIST; (b) Fashion-MNIST; and (c) CelebA datasets. While Poly-LSGAN learns meaningful representations (although visually sub-par compared to standard GANs) on MNIST and Fashion-MNIST, the generator fails to converge in all scenarios. *The poor performance of Poly-LSGAN can be attributed to training instability issues caused by the singularity of the monomial matrix* $\mathbf{B}$ *in solving for the optimal discriminator weights.*