Stabilizing GAN Training with Multiple Random Projections

Behnam Neyshabur Srinadh Bhojanapalli Ayan Chakrabarti
Toyota Technological Institute at Chicago
6045 S. Kenwood Ave., Chicago, IL 60637
{bneyshabur,srinadh,ayanc}@ttic.edu

Abstract

Training generative adversarial networks is unstable in high-dimensions when the true data distribution lies on a lower-dimensional manifold. The discriminator is then easily able to separate nearly all generated samples leaving the generator without meaningful gradients. We propose training a single generator simultaneously against an array of discriminators, each of which looks at a different random low-dimensional projection of the data. We show that individual discriminators then provide stable gradients to the generator, and that the generator learns to produce samples consistent with the full data distribution to satisfy all discriminators. We demonstrate the practical utility of this approach experimentally, and show that it is able to produce image samples with higher quality than traditional training with a single discriminator.

1 Introduction

Generative adversarial networks (GANs), introduced by [1], endow neural networks with the ability to express distributional outputs. The framework includes a generator network that is tasked with producing samples from some target distribution, given as input a (typically low dimensional) noise vector drawn from a simple known distribution, and possibly conditional side information. The generator learns to generate such samples, not by directly looking at the data, but through adversarial training with a discriminator network that seeks to differentiate real data from those generated by the generator. To satisfy the objective of "fooling" the discriminator, the generator eventually learns to produce samples with statistics that match those of real data.

In regression tasks where the true output is ambiguous, GANs provide a means to simply produce an output that is plausible (with a single sample), or to explicitly model that ambiguity (through multiple samples). In the latter case, they provide an attractive alternative to fitting distributions to parametric forms during training, and employing expensive sampling techniques at the test time. In particular, conditional variants of GANs have shown to be useful for tasks such as in-painting [2], and super-resolution [3]. Recently, [4] demonstrated that GANs can be used to produce plausible mappings between a variety of domains—including sketches and photographs, maps and aerial views, segmentation masks and images, *etc*. GANs have also found uses as a means of un-supervised learning, with latent noise vectors and hidden-layer activations of the discriminators proving to be useful features for various tasks [2, 5, 6].

Despite their success, training GANs to generate high-dimensional data (such as large images) is challenging. Adversarial training between the generator and discriminator involves optimizing a min-max objective. This is typically carried out by gradient-based updates to both networks, and the generator is prone to divergence and mode-collapse as the discriminator begins to successfully distinguish real data from generated samples with high confidence. Researchers have tried to address this instability and train better generators through several techniques. [7] proposed explicitly factorizing generating an image into a sequence of conditional generations of levels of a Laplacian

[2] Emily Denton, Sam Gross, and Rob Fergus. Semi-supervised learning with context-conditional generative adversarial networks. *arXiv:1611.06430 [cs.CV]*, 2016.

[3] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv:1609.04802 [cs.CV]*, 2016.

[4] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv:1611.07004 [cs.CV]*, 2016.

[5] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016.

[6] Alec Radford, Luke Metz, and Soumit Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[7] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, 2015.

[8] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016.

[9] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *NIPS Workshop on Adversarial Training*, 2016.

[10] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *NIPS*, 2016.

[11] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv:1701.07875 [stat.ML]*, 2017.

[12] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv:1609.03126 [cs.LG]*, 2016.

[13] Yaxing Wang, Lichao Zhang, and Joost van de Weijer. Ensembles of generative adversarial networks. *arXiv:1612.00991 [cs.CV]*, 2016.

[14] Ishan Durugkar, Ian Gemp, and Sridhar Mahadevan. Generative multi-adversarial networks. *arXiv:1611.01673 [cs.LG]*, 2016.

[15] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *ICML*, 1996.

[16] Robert A Jacobs. Methods for combining experts' probability assessments. *Neural computation*, 1995.

[17] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 2015.

[18] Sanjoy Dasgupta. Learning mixtures of gaussians. In *Foundations of Computer Science*. IEEE, 1999.

[19] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[20] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[22] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv:1011.3027 [math.PR]*, 2010.

Supplementary Material

A Proofs

Proof of Theorem 2.1. We first show that we can assume that the columns of the projection W are orthonormal. Since $W \in \mathbb{R}^{d\times m}$ is entry-wise Gaussian distributed, it has rank m with high probability. Then, there exists a square invertible matrix A such that $W' = AW$ where W' is orthonormal. In that case, $\text{Vol}(\text{supp}(\mathbb{P}_{W^T x}))/\text{Vol}(B^d_W) = \text{Vol}(\text{supp}(\mathbb{P}_{W'^T_x}))/\text{Vol}(B^d_{W'})$ because the numerator and denominator terms for both can be related by $\det(A)$ for the change of variables, which cancels out. Note that under this orthonormal assumption, $B^d_W = B^m$.

Next, we consider the case of an individual Gaussian distribution $\mathbb{P}_x = \mathcal{N}(x|\mu, \Sigma)$, and prove that the ratio of supports (defined with respect to a threshold ϵ) does not decrease with the projection. The expression for these ratios is given by:

$$\text{Vol}(\text{supp}(\mathbb{P}_x)) = \text{Vol}(B^d) \times \det(\Sigma) \times \left[\log \frac{1}{\epsilon^2} - d\log 2\pi - \log\det(\Sigma)\right]$$

$$\Rightarrow \frac{\text{Vol}(\text{supp}(\mathbb{P}_x))}{\text{Vol}(B^d)} = \det(\Sigma) \times \left[\log \frac{1}{\epsilon^2} - d\log 2\pi - \log\det(\Sigma)\right]. \tag{4}$$

$$\frac{\text{Vol}(\text{supp}(\mathbb{P}_{W^T x}))}{\text{Vol}(B^d_W)} = \det(W^T\Sigma W) \times \left[\log \frac{1}{\epsilon^2} - m\log 2\pi - \log\det(W^T\Sigma W)\right]. \tag{5}$$

For sufficiently small ϵ, the volume ratio of a single Gaussian will increase with projection if $\det(W^T\Sigma W) > \det(\Sigma)$. Note that all eigenvalues of $\Sigma \leq 1$, with at-least one eigenvalue strictly < 1 (since $\text{supp}(\mathbb{P}_x) \subset B^d$). First, we consider the case when Σ is not strictly positive definite and one of the eigenvalues is 0. Then, $\text{Vol}(\text{supp}(\mathbb{P}_x)) = 0$ and $\text{Vol}(\text{supp}(\mathbb{P}_{W^T x})) \geq 0$, *i.e.,* the volume ratio either stays the same or increases.

For the case when all eigenvalues are strictly positive, consider a co-ordinate transform where the first m co-ordinates of x correspond to the column vectors of W, such that

$$\Sigma = \begin{bmatrix} \Sigma_W & \Sigma_{WW'} \\ \Sigma^T_{WW'} & \Sigma_{W'} \end{bmatrix}, \tag{6}$$

where $\Sigma_W = W^T\Sigma W$. Then,

$$\begin{aligned} \det(\Sigma) &= \det(\Sigma_W)\det(\Sigma_{W'} - \Sigma^T_{WW'}\Sigma^{-1}_W\Sigma_{WW'}) \\ &\leq \det(\Sigma_W)\det(\Sigma_{W'}), \\ \Rightarrow \det(\Sigma_W) &\geq \det(\Sigma)/\det(\Sigma_{W'}). \end{aligned} \tag{7}$$

Note that $\det(\Sigma_{W'}) \leq 1$, since all eigenvalues of Σ are ≤ 1, with equality only when W is completely orthogonal to the single eigenvector whose eigenvalue is strictly < 1, which has probability zero under the distribution for W. Therefore, we have that $\det(\Sigma_{W'}) < 1$, and

$$\det(W^T\Sigma W) = \det(\Sigma_w) > \det(\Sigma). \tag{8}$$

The above result shows that the volume ratio of individual components never decrease, and *always* increase when their co-variance matrices are full rank (no zero eigenvalue). Now, we consider the case of the Gaussian mixture. Note that the volume ratio of the mixture equals the sum of the ratios of individual components, since the denominator $\text{Vol}(B^m)$ is the same, where the support volume in these ratios for component j is defined with respect to a threshold ϵ/τ_j. Also, note that since mixture distribution has non-zero volume, at least one of the Gaussian components must have all non-zero eigenvalues. Therefore, the volume ratios of $\mathbb{P}_x$ and $\mathbb{P}_{W^T x}$ are both sums of individual Gaussian component terms, and each term for $\mathbb{P}_{W^T x}$ is greater than or equal to the corresponding term for $\mathbb{P}_x$, and at least one term is strictly greater. Therefore, the support volume ratio of $\mathbb{P}_{W^T x}$ is strictly greater than that of $\mathbb{P}_x$.

Proof of Theorem 2.2. The proof of follows along the same steps as that of Theorem 1 in [1].

$$\begin{aligned} & V(D_k, G) \\ &= \mathbb{E}_{x\sim\mathbb{P}_x}[\log D_k(W_k^T x)] + \mathbb{E}_{x\sim\mathbb{P}_g}[\log(1 - D_k(W_k^T x))] \\ &= \mathbb{E}_{Y\sim\mathbb{P}_{W_k^T x}}[\log D_k(y)] + \mathbb{E}_{y\sim\mathbb{P}_{W_k^T g}}[\log(1 - D_k(y))]. \end{aligned} \tag{9}$$

For any point $y \in \text{supp}(\mathbb{P}_{W_k^T x}) \cup \text{supp}(\mathbb{P}_{W_k^T g})$, differentiating $V(D_k, G)$ w.r.t. D_k and setting to 0 gives us:

$$D_k(y) = \frac{\mathbb{P}_{W_k^T x}(y)}{\mathbb{P}_{W_k^T x}(y) + \mathbb{P}_{W_k^T g}(y)}. \tag{10}$$

Notice we can rewrite $V(D_k, G)$ as

$$\begin{aligned} V(D_k, G) = -2\log(2) + KL\left(\mathbb{P}_{W_k^T x} || \frac{\mathbb{P}_{W_k^T x} + \mathbb{P}_{W_k^T g}}{2}\right) \\ + KL\left(\mathbb{P}_{W_k^T g} || \frac{\mathbb{P}_{W_k^T x} + \mathbb{P}_{W_k^T g}}{2}\right). \end{aligned} \tag{11}$$

Here KL is the Kullback Leibler divergence, and it is easy to see that the above expression achieves the minimum value when $\mathbb{P}_{W_k^T x} = \mathbb{P}_{W_k^T g}$.

Proof of Theorem 2.3. We first prove this result for discrete distributions supported on a compact set $\mathcal{S}$ with γ points along each dimension. Let $\tilde{\mathbb{P}}$ denote such a discretization of a distribution $\mathbb{P}$.

Each of the marginal equation $\tilde{\mathbb{P}}_{W_k^T x} = \tilde{\mathbb{P}}_{W_k^T g}$ is equivalent to γ^m linear equations of the distribution $\tilde{\mathbb{P}}_x$ of the form, $\sum_{x:W_k^T x = y} \tilde{\mathbb{P}}_x(x) = \tilde{\mathbb{P}}_{W_k^T g}(y)$. Note that we have γ^d choices for x and γ^m choices for y. Let $A_k \in \mathbb{R}^{\gamma^m \times \gamma^d}$ denote the coefficient matrix $A_k \tilde{\mathbb{P}}_x = \tilde{\mathbb{P}}_{W_k^T g}$, such that $A_k(i,j) = 1$ if $W_k^T x_i = y_j$, and 0 otherwise.

The rows of A_k for different values of y_j are clearly orthogonal. Further, since different W_k are independent Gaussian matrices, rows of A_k corresponding to different W_k are linearly independent. In particular let $A \in \mathbb{R}^{\gamma^m K \times \gamma^d}$ denote the vertical concatenation of A_k. Then, A has full row rank of $\gamma^m \cdot K$ with probability $\geq 1 - c \cdot m \cdot e^{-d}$ [22]. Here c is some arbitrary positive constant.

Since $\tilde{\mathbb{P}}_x$ is a vector of dimension γ^d, that many linearly independent equations, uniquely determine it. Hence $\gamma^m \cdot K \geq \gamma^d$, guarantees that $\tilde{\mathbb{P}}_x = \tilde{\mathbb{P}}_g$.

Now we extend the results to the continuous setting. Without loss of generality, let the compact support $\mathcal{S}$ of the distributions be contained in a sphere of radius B. Let $\mathcal{N}_{\frac{\epsilon}{L}}$ be an $\frac{\epsilon}{L}$ net of $\mathcal{S}$, with γ^d points (see Lemma 5.2 in [22]), where $\gamma = 2B \cdot L/\epsilon$. Then for every point $x_1 \in \mathcal{S}$, there exists a $x_2 \in \mathcal{N}_{\frac{\epsilon}{L}}$ such that, $d(x_1, x_2) \leq \frac{\epsilon}{L}$.

Further for any x_1, x_2 with $d(x_1, x_2) \leq \frac{\epsilon}{L}$, from the Lipschitz assumption of the distributions we know that,

$$|\mathbb{P}_x(x_1) - \mathbb{P}_x(x_2)| \leq L \cdot \frac{\epsilon}{L} = \epsilon. \tag{12}$$

Finally, notice that the marginal constraints do not guarantee that the distributions $\tilde{\mathbb{P}}_{W_k^T x}$ and $\tilde{\mathbb{P}}_{W_k^T g}$ match exactly on the ϵ-net, but only that they are equal upto an additive factor of ϵ. Hence, combining this with equation 12 we get,

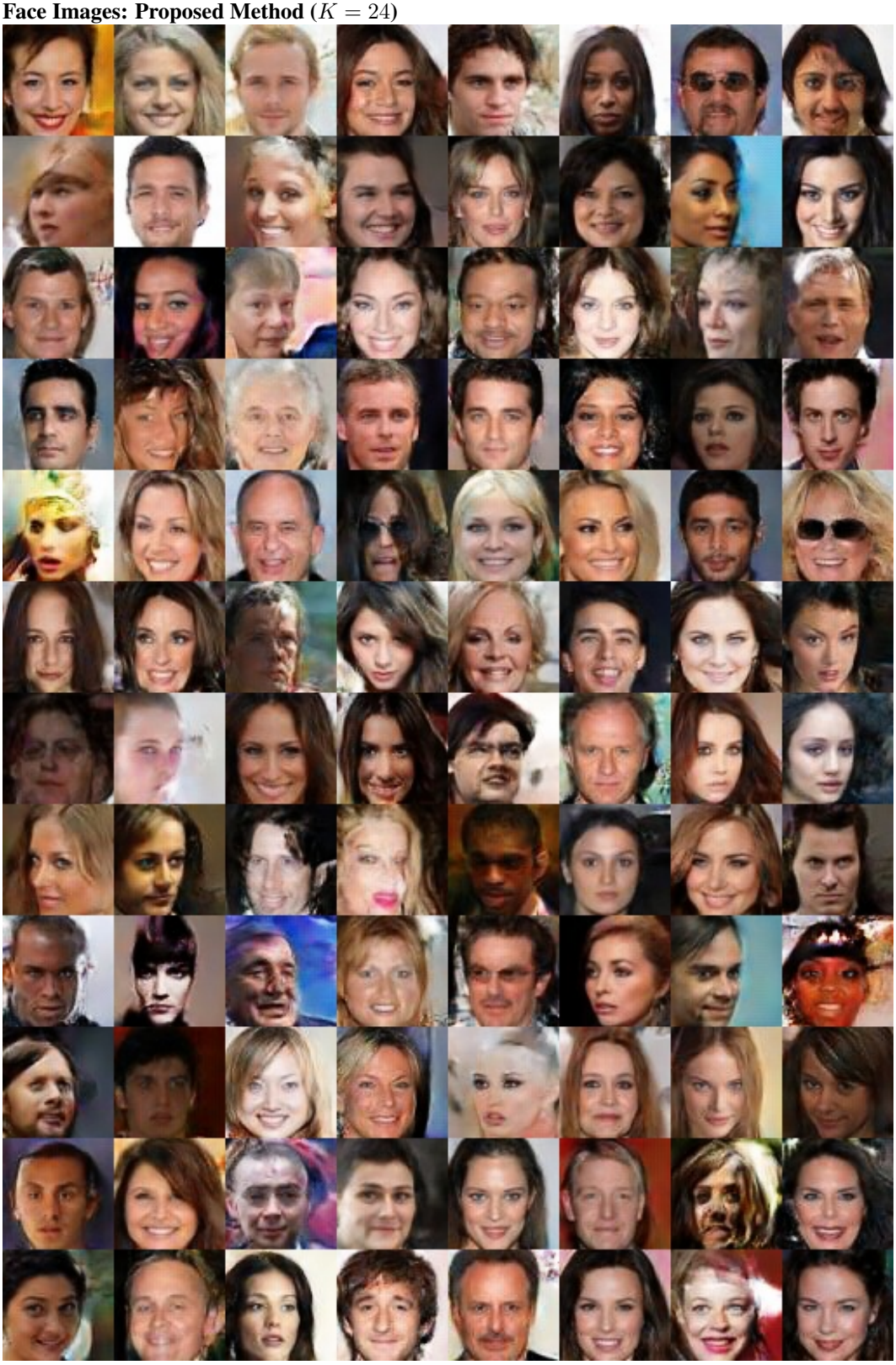
$$|\mathbb{P}_x(x) - \mathbb{P}_g(x)| \leq O(\epsilon),$$

for any x with probability $\geq 1 - c \cdot m \cdot K \cdot e^{-d}$.

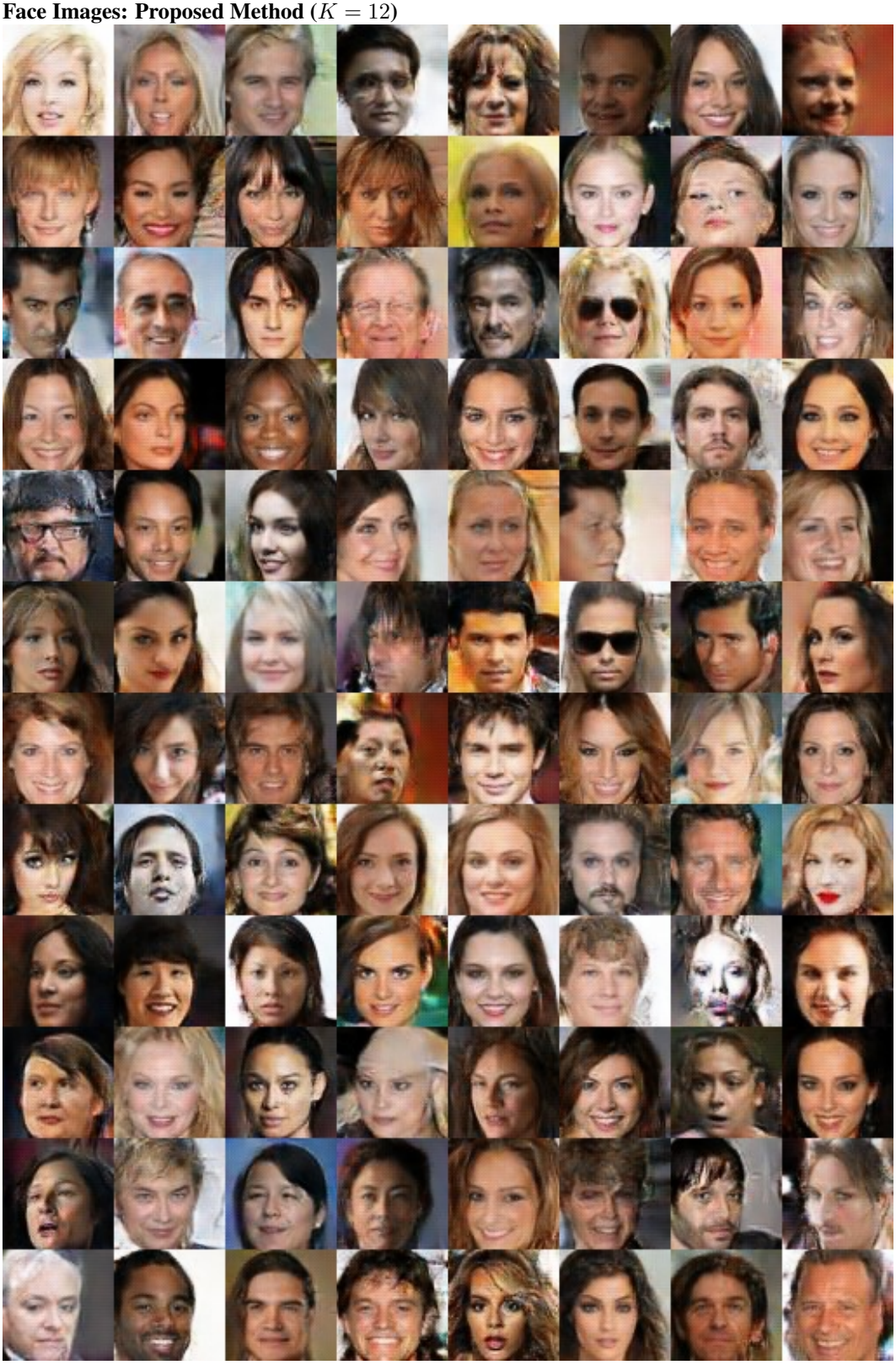
B Additional Experimental Results

Face Images: Proposed Method ($K = 48$)

Face Images: Proposed Method ($K = 24$)



Face Images: Proposed Method ($K = 12$)



Face Images: Traditional DC-GAN (Iter. 40k)

Face Images: Traditional DC-GAN (Iter. 100k)

Random Imagenet-Canine Images: Proposed Method