Indirect Adversarial Losses via an Intermediate Distribution for Training GANs – Supplementary Material

Rui Yang Duc Minh Vo Hideki Nakayama
{yang, vmduc}@nlab.ci.i.u-tokyo.ac.jp nakayama@ci.i.u-tokyo.ac.jp
The University of Tokyo, Japan

A. Proof of Lemma 1

Before the proof, two concepts are vital to be clarified. First, $f^*(\cdot)$ is the function of two functions: the IPM function conditioned on parameters in the discriminator function. In other words, $f^*(\cdot) = \mathrm{IPM}(D(\cdot))$, where $D(\cdot)$ is the discriminator function. Second, $D(\cdot)$ gains its Lipschitz continuity from spectral normalization[6] or gradient penalty[1]. In practice, the original MMD-GAN defined an auto-encoder as its discriminator. At the same time, other IPM-GANs normally use a binary classifier or an encoder to map input image space to an implicit feature space. Thus, output space is also bounded if input space is bounded and vice versa. Our methods followed the repulsive MMD-GAN's discriminator mapping real or fake images to a multi-dimensional feature space. Thus, $f^*(\cdot)$ is bounded, and the additive linear operation in Lemma 1 does not change this property. As a result, the sum of $f_q^*(r) + f_q^*(g)$ is still a real-valued bounded measurable function (assuming q as the intermediate distribution).

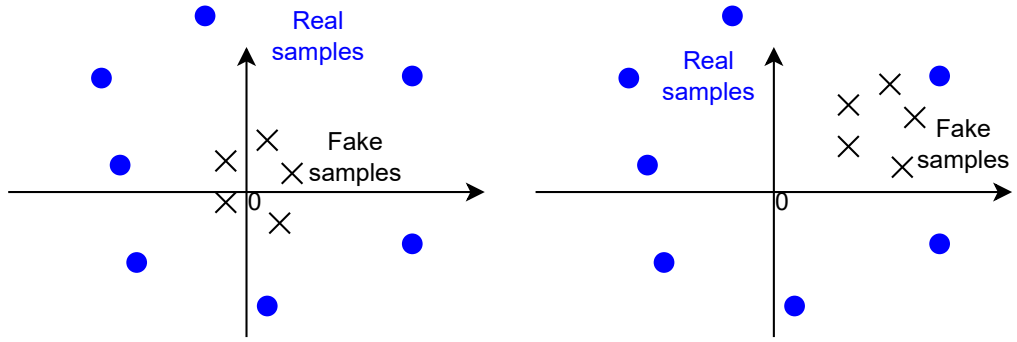
Proof. From [5] we can see if *inputs* to be measured in a topological space, Lemma 1 can work as the adversarial divergence. Hereon, the emphasized '*inputs*' are the ad hoc sets of the discriminator outputs, $D(\cdot)$. Traditionally defined 'source distribution' in our equations changes from the generated fake distribution to real or fake distributions. Meanwhile, traditionally defined 'target distribution' changes from the real distribution to the intermediate distribution, which is not in contrast to the requirement of a fixed target distribution as in [5]. Also, because the sum of the witness functions from real and fake sources is a real-valued bounded measurable function, such adversarial divergence can satisfy the definition of IPM-GANs[5, 7, 8].

$$
\begin{aligned}
&MMD_{q=\delta}(real, fake) = \\
&\inf_{\mathcal{G}} \sup_{f^* \in \mathcal{H}} \big| \mathbb{E}_{r \sim real}[||\mathcal{D}(r) - 0||_{\mathcal{H}}^2] \\
&+ \mathbb{E}_{g \sim fake}[||\mathcal{D}(g) - 0||_{\mathcal{H}}^2] \big|,
\end{aligned}
$$

$$
\begin{aligned}
&MMD_{q=\mathcal{N}}(real, fake) = \\
&\inf_{\mathcal{G}} \sup_{f^* \in \mathcal{H}} \big| \mathbb{E}_{r \sim real, N \sim \mathcal{N}}[||\mathcal{D}(r) - N||_{\mathcal{H}}^2] \\
&+ \mathbb{E}_{g \sim fake, N \sim \mathcal{N}}[||\mathcal{D}(g) - N||_{\mathcal{H}}^2] \big|.
\end{aligned}
$$

In MMD(δ) or MMD($\mathcal{N}$), real and fake inputs are mapped to the Reproducing Kernel Hilbert Space (RKHS, $\mathcal{H}$), which is the same as the original MMD-GAN[3], thus such mapping can satisfy the witness function [7] and never violate Lemma 1.

$$
\begin{aligned}
&KSD_{q=\mathcal{U}}(real, fake) = \\
&\inf_{\mathcal{G}} \sup_{f^* \in \mathcal{H}} \big| \mathbb{E}_{r, r' \sim real}[u_{\mathcal{U}}(\mathcal{D}(r), \mathcal{D}(r'))] \\
&+ \mathbb{E}_{g, g' \sim fake}[u_{\mathcal{U}}(\mathcal{D}(g), \mathcal{D}(g'))] \big|,
\end{aligned}
$$



(a) MMD(Rep.) (b) MMD(δ)

Figure 1: Discriminator outputs distributions of real and fake samples for MMD(Rep.) in (a) and MMD(δ) in (b).

$$KSD_{q=\mathcal{N}}(real, fake) = \inf_{\mathcal{G}} \sup_{f^* \in \mathcal{H}} \big| \mathbb{E}_{r,r' \sim real}[u_{\mathcal{N}}(\mathcal{D}(r), \mathcal{D}(r'))] + \mathbb{E}_{g,g' \sim fake}[u_{\mathcal{N}}(\mathcal{D}(g), \mathcal{D}(g'))] \big|.$$

Intermediate distribution in KSD-based [4] losses holds the implicit intermediate distribution, unlike the explicit random samples in MMD(δ) and MMD($\mathcal{N}$). In the KSD objective functions, because $u_q(\cdot)$ kernel also mapping $D(\cdot)$ inputs to RKHS, hence satisfying the witness function [7]. Thus our methods can inherit conclusions of the $weak$-* convergence as other types of IPM-GANs from the past literature[5]. □

B. Mismatching Problem

Based on the definition of KSD, the p.d.f. in the score function does not require an accurate normalization parameter. Such a non-parametric feature makes KSD very convenient in many aspects of machine learning topics, while it may cause the mismatching problem about the output scale at the initial steps of the adversarial training process. Here we found that adding a bound on the RBF kernel in the KSD objective functions to limit the output scale of the discriminator can avoid the mismatching problem efficiently. The hinged kernel is defined as follow [9]:

$$k_{RBF}\big(\mathcal{D}(g), \mathcal{D}(g')\big) = exp\big(\frac{1}{2\sigma^2} max(\|\mathcal{D}(g) - \mathcal{D}(g')\|^2, b_l)\big),$$
$$k_{RBF}\big(\mathcal{D}(r), \mathcal{D}(r')\big) = exp\big(\frac{1}{2\sigma^2} min(\|\mathcal{D}(r) - \mathcal{D}(r')\|^2, b_u)\big),$$

where σ is the bandwidth of the RBF kernel. b_l and b_u are the lower and upper bound. The hinged kernel was set as $b_l = 0.5, b_u = 2$ for KSD experiments and $b_l = 0.25, b_u = 4$ for MMD experiments.

C. Details from MMD(Rep.) to MMD(δ)

As introduced in Section 'Indirect Adversarial Losses' in the main paper, we rewrite the repulsive MMD discriminator loss as the zero-centered loss based on the original MMD function with regularization terms, which can achieve repulsiveness other than [9]. The meaning of MMD(δ) is the difference of two mathematically defined standard MMD distances. In contrast, MMD(Rep.) is the difference between two kernels, which does not have a clear mathematical meaning based on the standard MMD function.

Here we list the detailed equations:

$$
\begin{aligned}
&MMD(\delta) = MMD(p, \mathcal{O}) - MMD(q, \mathcal{O})) \\
&= \mathbb{E}_{x\sim p}[k(\mathcal{D}(x), \mathcal{D}(x'))] - 2 * \mathbb{E}_{x\sim p}[k(\mathcal{D}(x), 0)] \\
&- \mathbb{E}_{y\sim q}[k(\mathcal{D}(y), \mathcal{D}(y'))] + 2 * \mathbb{E}_{y\sim q}[k(\mathcal{D}(y), 0)] \\
&= MMD(Rep.) + exp(-\gamma * ||\mathcal{D}(x)||_2^2) \\
&- exp(-\gamma * ||\mathcal{D}(y)||_2^2) \\
&\propto MMD(Rep.) + ||\mathcal{D}(x)||_2 - ||\mathcal{D}(y)||_2.
\end{aligned}
$$

From the equation above, we can conclude the difference between our intermediate-based MMD(δ) and MMD(Rep.). In MMD(Rep.) [9], such loss only focuses on the intra-distance of real or fake distributions. In contrast, our zero-centered loss also limits the possibility of avoiding real outputs being too far away from zeros and stopping fake outputs from being too close to the zeros.

Here we introduce real and fake cases, respectively. For fake outputs, as shown in Fig 1^a, fake outputs may be concentrated towards the zero point (also shown in its official demonstration[1]), which can hardly bring too much information in numeric. While ours can avoid such a demerit as MMD(δ) maximizes the distance between fake outputs and the zero point. For real outputs, in MMD(Rep.), the center of real outputs in one mini-batch does not matter because such a loss only maximizes the intra-distance of real outputs. Thus, the first-order moment of the distribution of the real outputs was detached. While our MMD(δ) still keeps the same physical means as the original MMD (while toward zero), considering every order of moments. Besides, the E[k(D(x),D(x'))] term prevents real outputs become too small when moving such digits towards the zero in MMD(δ), which is the same function as achieving the repulsiveness in MMD(Rep.).

Our generator loss follows the same logic. MMD(Rep.) discriminator loss does not have a clear physical meaning, so it is hard to create a corresponding generator loss. Using original MMD loss directly can make it work while lacking the guarantee of convergence from past works. Our intermediate-distribution-based losses have physical meanings and can follow past literature, as shown in Appendix A.

All in all, the proposed intermediate-distribution-based MMD function achieved repulsiveness, which is different from the repulsive MMD-GAN [9], by avoiding using the negative MMD function for real sources (as in original MMD-GAN [3]).

[1] https://github.com/richardwth/MMD-GAN

D. Algorithm

Algorithm 1 KSD-GAN, our proposed algorithm.

Input: learning rates (α_g, α_d), batch size B, discriminator iterations n per generator step, training data distribution p, intermediate distribution q.
Parameter: $\mathcal{G}$ parameters θ, $\mathcal{D}$ parameters ϕ.

Initialize θ and ϕ ;
while θ has not converged **do**
  for j = 1,...,n **do**
    Sample real samples $\{r_i\}_{i=1}^B \sim p$ and i.i.d. noises $\{z_i\}_{i=1}^B \sim \mathcal{N}(0, 1)$
    Generate fake samples $\{g_i\}_{i=1}^B \leftarrow \mathcal{G}_\theta(z_1)...\mathcal{G}_\theta(z_B)$
    Compute $L_\mathcal{D}$
    Update $\phi' \leftarrow \phi - Adam(\alpha_d, \phi, \bigtriangledown_\phi L_\mathcal{D})$
  end for
  Sample i.i.d. noises $\{z_i\}_{i=1}^B \sim \mathcal{N}(0, 1)$
  Generate fake samples $\{g_i\}_{i=1}^B \leftarrow \mathcal{G}_\theta(z_1)...\mathcal{G}_\theta(z_B)$
  Compute $L_\mathcal{G}$
  Update $\theta' \leftarrow \theta - Adam(\alpha_g, \theta, \bigtriangledown_\theta L_\mathcal{G})$
end while

D.1. Complexity

Our KSD-based loss functions are computed with the outputs of the discriminator. Dislike WassersteinGAN or Vanilla GAN, our discriminator has multi-dimensional outputs, just like repulsive MMD-GAN. Assuming that the batch size in one generation step is B, the number of output dimensions in the discriminator is d. Thus the computational complexity is $O(d^2B)$ for our KSD-GAN and repulsive MMD-GAN (typically d varies from 16 to 64 and has a better performance as in [9]); the com-

plexity is $O(dB)$ for WassersteinGAN and Vanilla GAN (normally $d = 1$). Therefore, our KSD-GAN and repulsive MMD-GAN will have a considerably larger computational complexity in loss functions and the last layer of the discriminator. Finally, these will result in more than 10% longer training time than single-dimensional discriminators during each training iteration in our experiments.

E. Diversity

Here we show the diversity results with mean, max, min and variance statistics in box maps in 2.

F. Visualization on the output distribution

For testing the learning ability for inherent features, we trained MNIST [2] based on the DCGAN architecture without conditional labels in Fig. 4 and visualized outputs of the discriminator for real image inputs and in Fig. 3.

G. Extra Samples for Experiments

In our experiments, because CelebA dataset contains almost two times of images compared to another 64x64 resolution dataset Mini-ImageNet, we generate CelebA images within 50k training iterations to keep a reasonable total training time consumption. In this case, repulsive MMD-GAN cannot obtain convergence results while others can achieve reasonable FID scores. If we feed the model within enough training iterations (200k), all models can converge and obtain further improvements on FID scores of KSD-GAN, as shown below. We show random samples for CIFAR10, CIFAR100, and Mini-ImageNet and CelebA experiments as supplements in Fig. 5,6,7.

References

[1] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5769–5779, 2017.

[2] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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

[4] Qiang Liu, Jason Lee, and Michael Jordan. A kernelized stein discrepancy for goodness-of-fit tests. In *International conference on machine learning*, pages 276–284. PMLR, 2016.

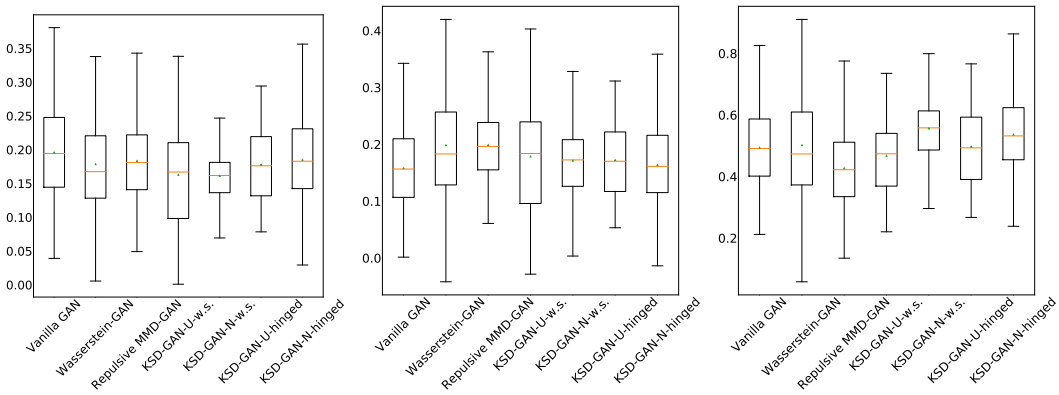
[5] Shuang Liu, Olivier Bousquet, and Kamalika Chaudhuri. Approximation and convergence properties of generative adversarial learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5551–5559, 2017.

[6] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for gans. In *International Conference on Learning Representations*, 2018.

[7] Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, pages 429–443, 1997.

[8] Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert RG Lanckriet. On integral probability metrics,\phi-divergences and binary classification. *arXiv preprint arXiv:0901.2698*, 2009.

[9] Wei Wang, Yuan Sun, and Saman Halgamuge. Improving mmd-gan training with repulsive loss function. In *International Conference on Learning Representations*, 2019.

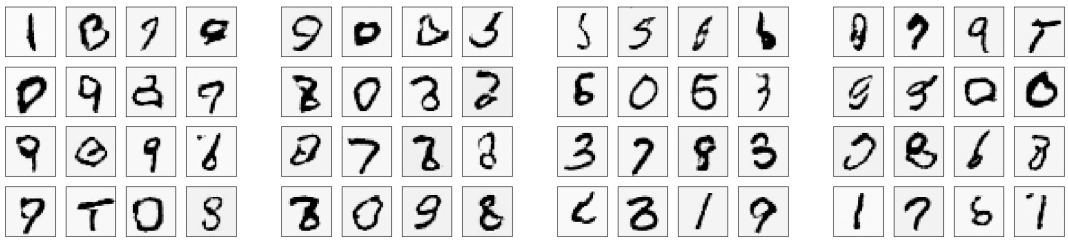


(a) Cifar10 (b) Cifar100 (c) Tiny-Imagenet

Figure 2: LPIPS (higher is better) box-plots among different classes in three datasets. For every dataset, we first calculated the LPIPS score for each class and then drew the box-plot among all the classes and repeat steps for different settings. Green triangles indicate mean values and yellow lines are median numbers.

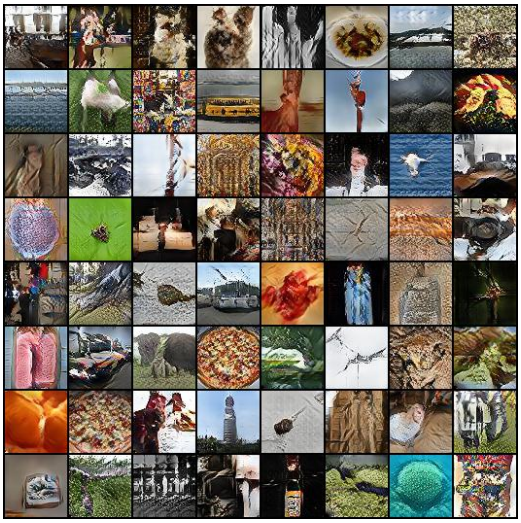
(a) Vanilla(JS) (b) Wasserstein (c) MMD(Rep.) (d) KSD($\mathcal{U}$) (e) KSD($\mathcal{N}$)

Figure 3: MNIST visualization maps for discriminator outputs from random samples. We first train an unconditional DCGAN and then visualize the output for the discriminator for real image inputs and label them with colors. Our method can acquire the inherent relations among classes.

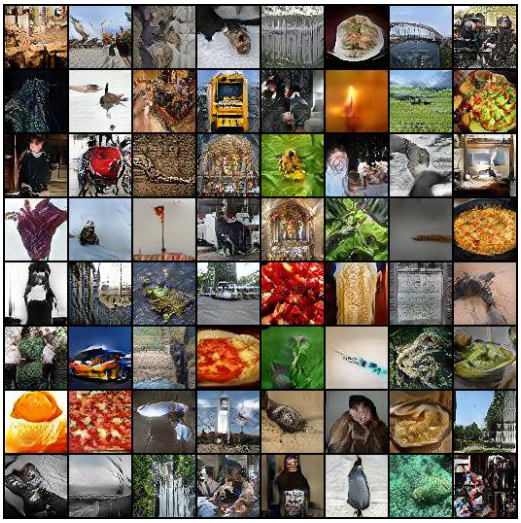


(a) KSD(U) (b) KSD(N) (c) Wasserstein (d) Repulsive-MMD

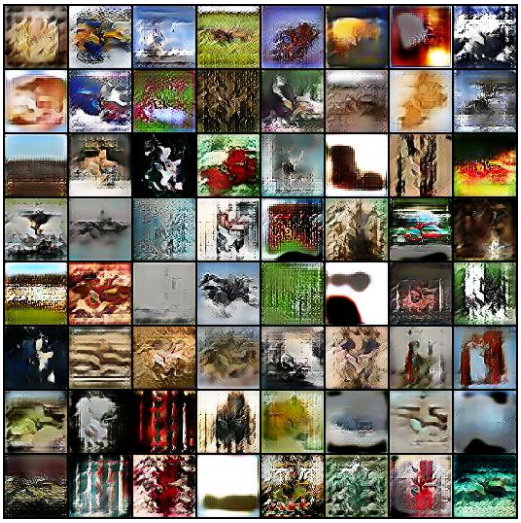
Figure 4: MNIST random samples on DCGAN.



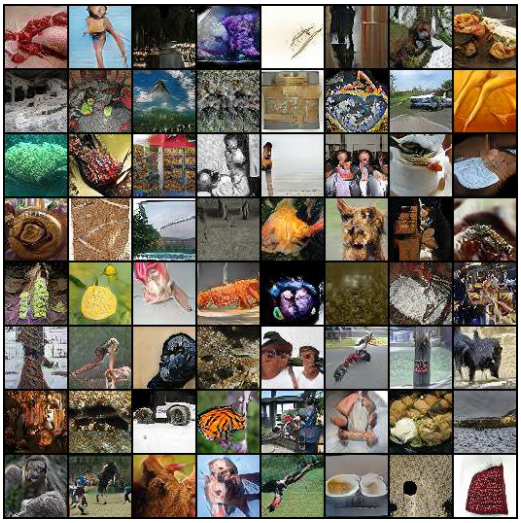
(a) Vanilla(JS) (IS:10.48)



(b) Was. (IS:12.67)



(c) Rep. (IS:6.08)



(d) KSD(U) (IS:12.71)

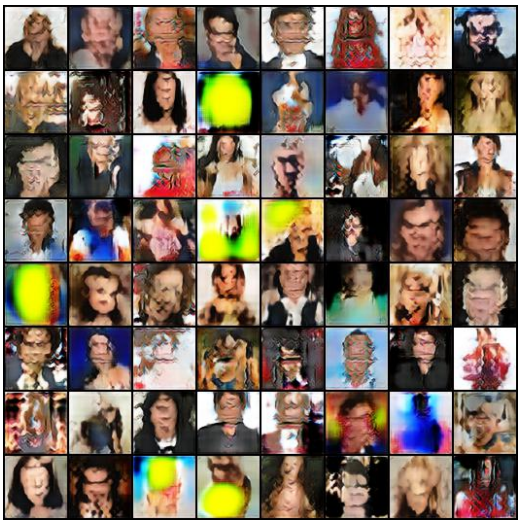
Figure 5: Random samples of Mini-ImageNet experiments.

(a) KSD (50k)

(b) KSD (FID:**3.63)**

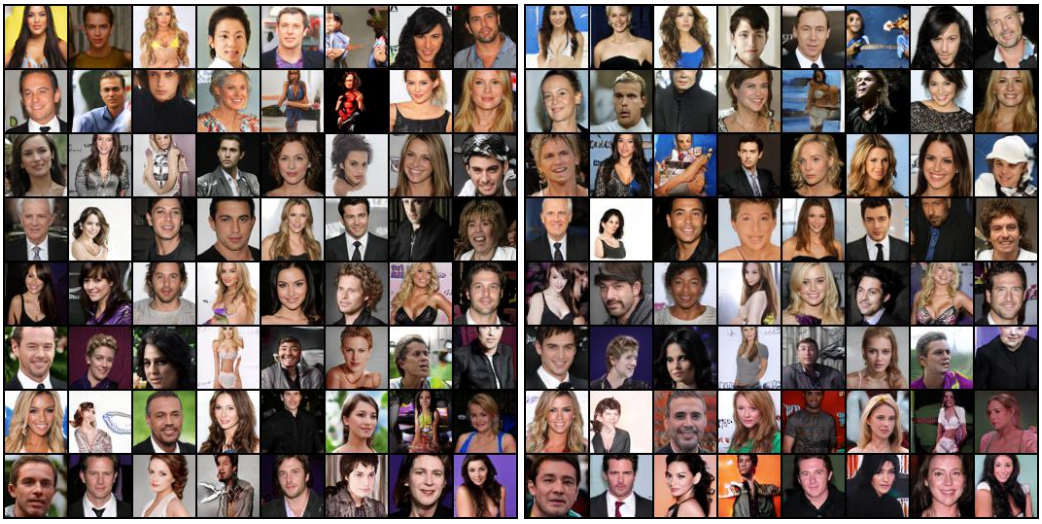
(c) Was. (50k)

(d) Was. (FID:7.13)



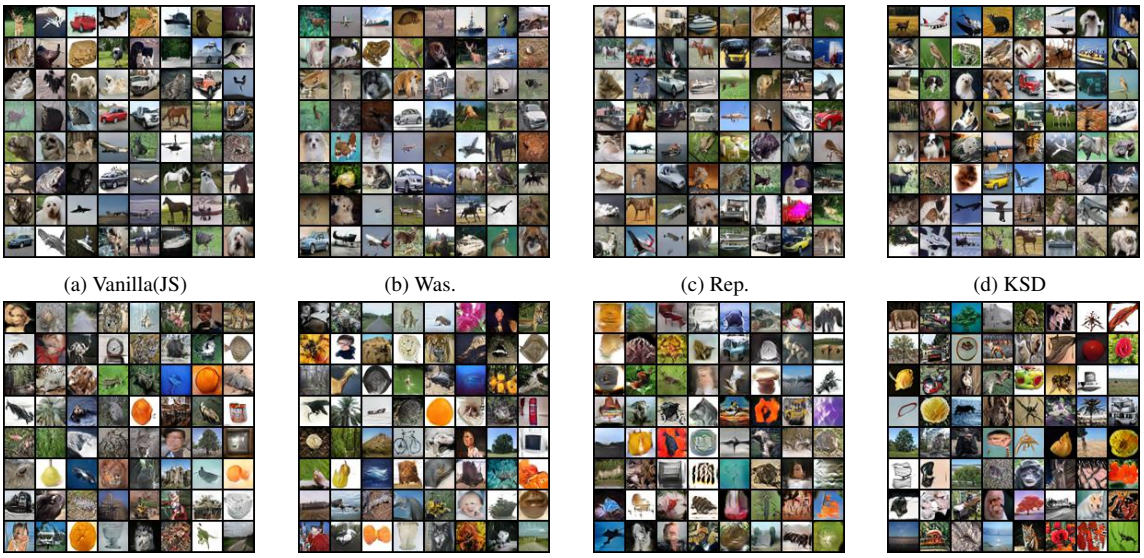
(e) Rep. (50k)

(f) Rep. (FID:12.78)



(g) Vanilla(JS) (50k) (h) Vanilla(JS)(FID:8.53)

Figure 6: Generation samples of CelebA after 50k and 200k training iterations.



(a) Vanilla(JS) (b) Was. (c) Rep. (d) KSD

(e) Vanilla(JS) (f) Was. (g) Rep. (h) KSD

Figure 7: Demonstrations of random samples from vanilla GAN, Wasserstein-GAN, Repulsive MMD-GAN, and KSD-GAN on Cifar10 (a,b,c,d) and Cifar100 (e,f,g,h). For KSD samples, we use the best-performed settings in Table 3 in main content.