MCL-GAN: GENERATIVE ADVERSARIAL NETWORKS WITH MULTIPLE SPECIALIZED DISCRIMINATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a generative adversarial network with multiple discriminators, which collaborate to represent a real dataset more effectively. This approach facilitates learning a generator consistent with the underlying data distribution based on real images and thus mitigates the chronic mode collapse problem. From the inspiration of multiple choice learning, we guide each discriminator to have expertise in the subset of the entire data and allow the generator to find reasonable correspondences between the latent and real data spaces automatically without the extra supervision for training examples. Despite the use of multiple discriminators, the backbone networks are shared across the discriminators and the increase of training cost is marginal. We demonstrate the effectiveness of our algorithm using multiple evaluation metrics in the standard datasets for diverse tasks.

1 INTRODUCTION

Generative models learn to represent a probability distribution of data. With recent advances of deep generative models, Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) and Variational Autoencoders (VAEs) (Kingma & Welling, 2014) have shown impressive achievements in unconditional generation of high-dimensional realistic images as well as various conditional generation tasks including image-to-image translation (Zhu et al., 2017b; Lee et al., 2018; Zhu et al., 2017a), image inpainting (Yeh et al., 2017), image super-resolution (Ledig et al., 2017), *etc*.

GANs have received a lot of attention due to their interesting framework of minimax games, where two agents, a generator and a discriminator, compete against each other. Specifically, a discriminator distinguishes whether a sample comes from the real dataset or the generator while the generator attempts to deceive the discriminator. In theory, the generator learns the real data distribution by reaching an equilibrium point of the minimax game. It is known that GANs produce acute, high quality images compared to VAEs. However, in practice, the alternating training procedure does not guarantee the convergence to the optimal solution and often experiences mode collapsing, failing to cover the multiple modes of real data or, even worse, reaching at trivial solutions.

This paper focuses on the mode collapse problem in training GANs. Our main idea is adopting multiple collaborating discriminators. Each discriminator is learned to specialize in a subset of reference data space, which is identified automatically via the training procedure, so the ensemble of discriminators provide not only the differentiation of fake data, but also more accurate predictions over the clusters of real data. In this respect, a generator is encouraged to produce diverse modes that deceive a set of discrimantors. We employ Multiple Choice Learning (MCL) to learn multiple discriminators that are trained on a subset of training data as illustrated in Figure 1. The generator is updated via a set of expert models, each of which is associated with a subset of the true and generated examples closest to the expert. We call the proposed approach based on a single generator and multiple discriminators MCL-GAN, which is optimized by the standard objective of GAN combined with the objective for MCL in the discriminator side.

There are several GAN literatures that employ multiple discriminators (Nguyen et al., 2017; Durugkar et al., 2017; Albuquerque et al., 2019). Among them, GMAN (Durugkar et al., 2017) is closely related with our approach in the sense that it utilizes the ensemble prediction of discriminators. It explores multi-discriminator extensions of GANs with diverse versions of the aggregated prediction of discriminators—from a harsh trainer to a lenient teacher with a softened criteria. Meanwhile, there are significant differences in the method of ensembling from our approach. While GMAN focuses on the loss to the generator with parallel learning of discriminators, our strategy takes care of the specialization of each discriminator for more informative feedbacks to the generator.

The training algorithm of the proposed method is inspired by Multiple Choice Learning (Lee et al., 2016), which is known to be effective in learning specialized models with high oracle accuracy in recognition tasks. Encouraged by this benefit, Chen & Koltun (2017); Mun et al. (2018); Firman et al. (2018); Li et al. (2019) apply MCL or its variations (Lee et al., 2017; Tian et al., 2019) to produce diverse and accurate outputs in several applications. For instance, Mun et al. (2018) propose MCL-KD framework to come up with the visual question answering (VQA) systems based on multiple models that are specialized in different types of visual reasonings. Li et al. (2019) apply MCL to a conditional generative model for synthesizing diverse image from semantic layouts. DiverseNet (Firman et al., 2018) introduces the control parameter as an input that diversifies the outputs of networks with an MCL loss by making each control parameter ally with a different mode of data. While these works generate multiple outputs explicitly and select them at inference time, our approach adopts a unique strategy for diversifying the mode, learning to branch the decision of discriminators.

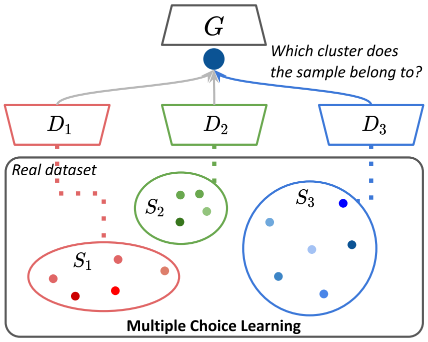


Figure 1: The main idea of MCL-GAN. Each discriminator D_m is trained to specialize in the cluster S_m of the real dataset. The mapping between D_m and S_m is obtained automatically by MCL.

The proposed method takes an advantage of MCL techniques into unconditional generative models, which has not been explored before. No supervision such as class labels or other conditions are assumed unlike the aforementioned works. Our main contributions are summarized as follows:

- We propose a single-generator multi-discriminator GAN training algorithm to alleviate the mode collapse problem. Our approach provides simple yet effective updating rules based on MCL to achieve the goal.
- We present a balanced discriminator assignment strategy to facilitate the robust convergence of models and preserve the multi-modality of training data, where the number of the discriminators is determined adaptively.
- The proposed method is applicable to many GAN variants since there is no constraint on the network architectures or the loss functions. Our method requires a small additional overhead and trains the model with computational efficiency via feature sharing in the discriminators.
- We experimentally show the competence of our method in terms of the generated image quality and the behavior of the networks.

2 Related work

There exist a lot of GAN approaches that address the mode collapse problem for output diversity. This section discusses the recent progress related to the issue briefly.

2.1 Handling mode collapse for diversity

Many variations of GANs propose either novel metrics for the discriminator loss or better alternatives of the discriminator design. For example, LSGAN (Mao et al., 2017) substitutes the least square function for the binary cross-entropy function as the discriminator loss. WGAN (Arjovsky et al., 2017) introduces a critic function based on the Earth-Mover's distance rather than a binary classifier, and WGAN-GP (Gulrajani et al., 2017) improves WGAN by adding a gradient penalty term. PacGAN (Lin et al., 2018) augments the discriminator's input by packing samples for a single label. EBGAN (Zhao et al., 2017) models the discriminator as an energy function, which is, in effect, implemented by the reconstruction loss of the autoencoder. BiGAN (Donahue et al., 2017), ALI (Dumoulin et al., 2017), VEEGAN (Srivastava et al., 2017), Inclusive GAN (Yu et al., 2020) also learn reconstruction networks. In particular, VEEGAN (Srivastava et al., 2017) autoencodes the latent vectors to learn the inverse function of the generator and map both the true and generated data to the latent distribution, *i.e.* a Gaussian. Inclusive GAN (Yu et al., 2020) learns a generator by matching between real and fake examples in the feature space.

The mode collapse and diversity issue of generated outputs has been addressed explicitly in (Liu et al., 2019; Yang et al., 2018; Mao et al., 2019). They formulate the diversity metrics that encourage the mode exploration of the generators and derive the loss function using the metrics. To be specific, Liu et al. (2019) measure normalized pairwise distances between the latent vectors and between their corresponding outputs, which are employed as a diversity loss to optimize the generator.

2.2 GAN WITH MULTIPLE GENERATORS

Another line of research is the integration of multiple generators (Tolstikhin et al., 2017; Ghosh et al., 2018; Hoang et al., 2018; Park et al., 2018). This approach represents the data distribution with a mixture model enforcing each generator to cover a portion of the whole data space. It is naturally expected that mixture models approximate true distributions better than a single model especially in high-dimensional spaces with multiple modes.

MAD-GAN (Ghosh et al., 2018) introduce an augmented classifier as a discriminator, which predicts whether the sample is real and which generator the sample is drawn from, to encourage individual generators to learn distinctive modes. MGAN (Hoang et al., 2018) has the similar strategies to MAD-GAN, but constructs a separate branch in the discriminator to perform the two tasks. MEGAN (Park et al., 2018) adopts a gating network that produces a one-hot vector to select the generator creating the best example. P2GAN (Trung Le et al., 2019) sequentially adds a new generator to cover the missing modes of the real data.

2.3 GAN WITH MULTIPLE DISCRIMINATORS

Multiple discriminators are often employed to improve the performance of a single generator (Nguyen et al., 2017; Durugkar et al., 2017; Albuquerque et al., 2019; Doan et al., 2019). D2GAN (Nguyen et al., 2017) conducts a three player minimax game, where two discriminators are trained for the completely opposite objectives, minimizing Kullback-Leibler (KL) divergence and the inverse KL divergence between the true and generated data distributions. The balancing of two losses plays a role for seeking desirable and diverse modes at the same time. Albuquerque et al. (2019) propose a general multi-objective optimization framework in the scenario with multiple discriminators. They present the hypervolume maximization algorithm to obtain weighed gradients. Neyshabur et al. (2017) train a GAN based on multiple projections. Each discriminator makes a decision for the random low-dimensional projection of a sample to address the instability of GAN training in high-dimensions.

GMAN (Durugkar et al., 2017) presents diverse aggregation methods of multiple discriminators, where both hard and soft discriminator selection strategies are studied. Note that all the existing approaches learn the multiple discriminators independently and they may have strong correlations, which may not be appropriate for diversifying the generated samples. Our approach, however, assigns each sample to the best-suited discriminator through the interactions among the discriminators, and, consequently, each discriminator becomes the expert model for the assigned examples.

3 MULTIPLE CHOICE LEARNING

We present the main idea of MCL (Guzman-Rivera et al., 2012) and its extensions briefly. Given a training dataset with N samples, $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, M models, $\{f_m\}_{m=1}^M$ and a task-specific loss function, $\ell(\cdot, \cdot)$, MCL minimizes the following oracle loss:

$$\mathcal{L}_{\text{MCL}}(\mathcal{D}) = \sum_{i=1}^{N} \min_{m} \ell(y_i, f_m(\mathbf{x}_i)). \tag{1}$$

In other words, only the model with the smallest error out of M candidates is selected for each example. This optimization process makes each model f_m become an expert for a subset of $\mathcal{D}$, thus leads to forming a natural cluster in $\mathcal{D}$.

A weakness of MCL is the possible mistakes caused by the overconfidence issues. If non-specialized models make wrong predictions with high confidences in the score aggregation process, the average scores are misleading and the ensemble model may result in poor quality outputs. To alleviate the limitation, Confident Multiple Choice Learning (CMCL) (Lee et al., 2017) adopts a confident oracle loss that enforces the predictions of a non-specialized model to be uniformly distributed using KL divergence, denoted by D_{KL}. Assuming that f_m predicts the output distribution given data point x, *i.e.*, $P_m(y|x)$, the modified loss is modified as

$$\mathcal{L}_{\text{CMCL}}(\mathcal{D}) = \sum_{i=1}^{N}\sum_{m=1}^{M} v_{i,m}\ell(y_i, P_m(y|\mathbf{x}_i)) + \beta(1 - v_{i,m})D_{\text{KL}}(\mathcal{U}(y)\|P_m(y|\mathbf{x}_i)), \tag{2}$$

where $\mathcal{U}(y)$ is the uniform distribution and the flag variable $v_{i,m} \in \{0, 1\}$ allows the choices of the specialized models. Note that, if $\sum_{m=1}^{M} v_{i,m} = k$ ($k < M$), each example is assigned to k models.

4 MCL-GAN

We describe our GAN structure with a generator $G(\cdot;\theta)$ and M discriminators $\{D_m(\cdot;\phi_m)\}_{m=1}^{M}$ extended from the standard GAN. Let p_z and p_d be the distributions of the latent space and real data space, respectively. Given $\mathbf{z} \sim p_z$, the generator produces a sample $\tilde{\mathbf{x}} = G(z;\theta)$ and M predictions are made by the discriminators for each real example $\mathbf{x} \sim p_d$ and fake sample $\tilde{\mathbf{x}}$. Each prediction, $D_m(\mathbf{x};\phi_m)$, ranges in $[0, 1]$ and represents the probability that $\mathbf{x}$ belongs to the true data distribution.

4.1 Expert training

Assuming that we draw N_d real data and generate N_g examples in each training batch, denoted by $\mathbf{x}$ and $\tilde{\mathbf{x}}$, respectively, each network is trained as follows.

Discriminators Expert discriminators are the ones that predict the highest scores for each sample. With the indicator variable $v_{i,m}$ for sample $\mathbf{x}_i$, the discriminators are trained to minimize the following loss function:

$$\mathcal{L}_{\text{e}}(\mathbf{x}) = -\sum_{i=1}^{N_d}\sum_{m=1}^{M} v_{i,m}\log(D_m(\mathbf{x}_i;\phi_m)), \tag{3}$$

where we choose k experts out of M discriminators for each example, *i.e.*, $\sum_{m=1}^{M} v_{i,m} = k$. In the case of a fake sample, all discriminators have to identify it correctly. Thus the following standard loss is added to equation 3:

$$\mathcal{L}_{\text{e}}(\tilde{\mathbf{x}}) = -\sum_{j=1}^{N_g}\sum_{m=1}^{M}\log(1 - D_m(G(\mathbf{z}_j);\phi_m)). \tag{4}$$

Generator We train the generator with respect to the gradients given by the expert models to encourage the generator to find the closest mode given $\mathbf{z}$. With another indicator variable $u_{j,m}$ for $\mathbf{z}_j$, the expert loss for the generator is given by

$$\mathcal{L}_{\text{e}}(\tilde{\mathbf{x}}) = \sum_{j=1}^{N_g}\sum_{m=1}^{M} u_{j,m}\log(1 - D_m(G(\mathbf{z}_j;\theta));\phi_m), \tag{5}$$

where $\sum_{m=1}^{M} u_{j,m} = k$.

4.2 Non-expert training

The non-expert discriminators should not be over-confident to real example while it is desirable to produce higher scores for real samples than fake ones. For this requirement, we give a uniform soft label, *e.g.*, $y = [0.5, 0.5]$ for non-expert discriminators and regularize them with some weight. To be precise, we obtain the following non-expert loss term corresponding to equation 3:

$$\mathcal{L}_{\text{ne}}(\mathbf{x}) = \sum_{i=1}^{N_d}\sum_{m=1}^{M}(1 - v_{i,m})\ell_{\text{ce}}(D_m(\mathbf{x}_i), y), \tag{6}$$

with the same $v_{i,m}$ defined in equation 3 and $\ell_{\text{ce}}(\cdot, y)$ is the cross-entropy loss function given a target label y. The other counterpart for equation 5 is derived similarly as

$$\mathcal{L}_{\text{ne}}(\tilde{\mathbf{x}}) = \sum_{j=1}^{N_g} \sum_{m=1}^{M} (1 - u_{j,m}) \ell_{\text{ce}}(D_m(G(\mathbf{z}_j)), y). \tag{7}$$

The non-expert model training is effective to handle the overconfidence issue, but the model may still suffer from the data deficiency problem of the standard MCL framework because each discriminator can see only a subset of the whole dataset. To ameliorate this limitation, our discriminators share the parameters of all layers for feature extraction while branching the last layer only. This implementation is also sensible in that the discriminators partially have the same objective to distinguish the fake examples. The common representations of all real samples are likely to be learned in the earlier layers despite being clustered in the different subsets whereas the critical information for the high-level classification is often found in the last layer. Moreover, the number of training parameters and training time are saved sigificantly while taking advantage of ensemble learning.

4.3 Balanced assignment of discriminators

On top of the adversarial losses, we introduce another loss for balanced updates of discriminators. As our training does not include any supervision for the specialized factor for certain discriminator, *e.g.*, class labels or feature embeddings, it may be difficult to reasonably distribute real samples to expert models from the beginning. Since the abilities of individual discriminators are severely off-balanced, they are highly prone to assign all samples to few specific models. Especially at an early phase of training, the model's capability is more sensitive to the number of updates in the discriminators.

To tackle this challenge, we propose another loss, namely a balance loss, that gives discriminators balanced chances to be updated. We let the selection of expert discriminators approximately follow a categorical distribution with a parameter $\boldsymbol{\mu} = [\mu_1, \dots, \mu_M]$. Then the loss is computed by the KL divergence of the probability distribution of discriminators for being selected as experts from $\boldsymbol{\mu}$. To obtain the probability for discriminator selection, we apply the `softmax` function to the vector of M predictions of discriminators—more precisely, logits before `sigmoid` function—for each example since the discriminator with the highest score is guaranteed to be chosen as an expert. We average these probability vectors over the training batch. *i.e.*, $\mathbf{q} = \frac{1}{N_d} \sum_{i=1}^{N_d} \mathtt{s}([D_1(\mathbf{x}_i), \dots, D_M(\mathbf{x}_i)]; \tau)$, where $\mathtt{s}(\cdot; \tau)$ denotes a vector-valued `softmax` function with temperature τ given an input vector. To sum up, the balance loss is given by

$$\mathcal{L}_{\text{bal}}(\mathbf{x}) = D_{\text{KL}}(\boldsymbol{\mu} \| \mathbf{q}). \tag{8}$$

In practice, we set $\mu_m = \frac{1}{M}, \forall m$ to update the discriminators evenly, which is because the true distribution is unavailable. This assumption may not be congruent to the real distribution of the dataset and excessively forced assignment would not result in an optimal clustering for specialization. We, therefore, decrease the weight for the balance loss gradually during training. Eventually, each example will be naturally assigned to its best model with a very small weight of the balance loss. This adjustment helps stabilize training and naturally cluster the reference data. Note that the models are balanced within a few epochs and the weight reduction helps generate higher quality samples.

Likewise, a small enforcement on the distribution of the generator's output facilitates balanced generation when the statistics of generated samples are skewed. For this case, we use the distribution of the discriminators' assignments instead of arbitrarily chosen $\boldsymbol{\mu}$, *i.e.*,

$$\mathcal{L}_{\text{bal}}(\tilde{\mathbf{x}}) = D_{\text{KL}}(\mathbf{q} \| \mathbf{o}). \tag{9}$$

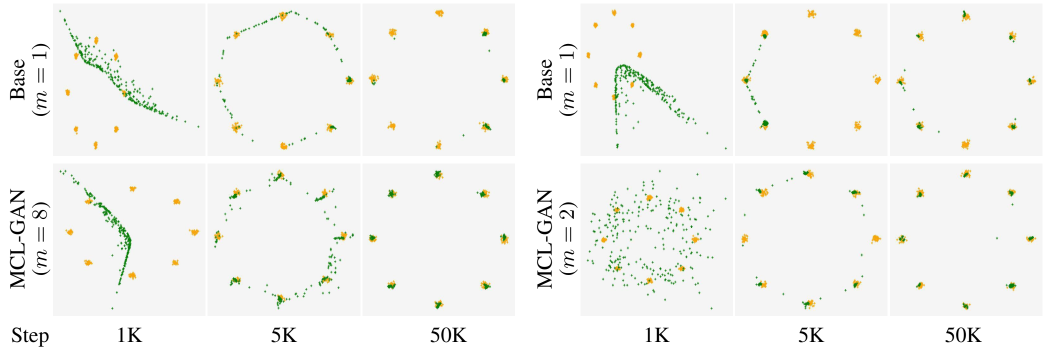
where $\mathbf{o} = \frac{1}{N_g} \sum_{j=1}^{N_g} \mathtt{s}([D_1(G(\mathbf{z}_j)), \dots, D_M(G(\mathbf{z}_j))]; \tau)$.

4.4 Total loss

Altogether, the total loss is summarized as follows:

$$\mathcal{L} = \mathcal{L}_{\text{e}} + \alpha \mathcal{L}_{\text{ne}} + \beta \mathcal{L}_{\text{bal}}, \tag{10}$$

where β is different for discriminators and generator. Although we describe the loss functions based on the standard GAN, it is applicable to other GAN formulations with different adversarial losses.



(a) Standard GAN loss

(b) Hinge loss

Figure 2: Snapshots of 256 random samples drawn from the generators of the baseline and MCL-GAN. Data sampled from the true distribution are in orange while the generated ones are in green.

4.5 Choice of number of discriminators

A remaining concern is that we need to find the optimal number of discriminators while such information is not available in general as in many clustering tasks. If the number of discriminators is much larger than the optimal one, it is more desirable to focus on training a subset of discriminators than dividing the dataset into many minor clusters forcefully.

To ease this issue, we employ L_1 regularization on the outputs of the discriminators, which encourages the sparsity of the discriminator selection and leads to more desirable clustering results. Hence, even in the case that we are given an excessively large number of discriminators, our algorithm converges at good points by using a small number of discriminators in practice. It is true that this strategy may not always lead to the optimal number of discriminators and has conflict with the balance loss in equation 8. However, the balance loss fades away as training goes, and our model identifies a proper number of clusters by deactivating a subset of discriminators. This sparsity loss may be useful when we learn on the examples drawn from unknown distributions.

5 Experiments

5.1 Synthetic data

We first perform toy experiments to verify the main idea of the proposed method intuitively. We consider a 2D mixture of 8 isotropic Gaussians whose centers are aligned on a circle with a radius $\sqrt{2}$ while their standard deviation in each dimension is set to 0.05. We employ 8 discriminators for training with the standard GAN loss while utilizing 2 discriminators for the model with Hinge loss (Lim & Ye, 2017). We choose one expert discriminator for each sample ($k = 1$) in all experiments.

Figure 2 illustrates the snapshots of random samples through iterations generated by the baselines and MCL-GANs. Unlike the base models ($m = 1$) fail to cover all 8 modes, MCL-GANs learn to identify diverse modes quickly and produce the samples at all modes eventually.

Appendix A presents that, when MCL-GANs are learned with an excessive number of discriminators, *e.g.*, $m = 20$, they mostly utilize 8 or 16 expert discriminators in a wide range of weight for the L_1 loss. This implies that MCL-GAN covers all the modes effectively and robustly.

5.2 Unconditional GAN on image dataset

We run the unconditional GAN experiment on four distinct datasets including MNIST (LeCun & Cortes, 2010), Fashion-MNIST (Xiao et al., 2017), CIFAR-10 (Krizhevsky et al., 2009) and CelebA (Liu et al., 2015), where two types of network architectures are employed—DCGAN (Radford et al., 2016) and StyleGAN2 (Karras et al., 2020). The images are resized to 32×32 except for CelebA dataset: 64×64 for DCGAN and 128×128 for StyleGAN2 experiment. For StyleGAN2 experiments on CelebA, we use the first and the last 30K images from the align&cropped version for a train and a

Table 1: Precision and recall scores from PRD curves on MNIST, Fashion-MNIST and CelebA datasets with the DCGAN architecture.

Loss	Method	m	MNIST		Fashion-MNIST		CelebA	
			Rec.↑	Prec.↑	Rec.↑	Prec.↑	Rec.↑	Prec.↑
GAN	Base (Radford et al., 2016)	1	0.896	0.778	0.936	0.900	0.834	0.839
	GMAN (Durugkar et al., 2017)	5	0.968	0.976	0.909	0.955	0.888	0.873
	GMAN (Durugkar et al., 2017)	10	0.964	0.977	0.928	0.946	0.921	0.923
	MCL-GAN	5	0.985	0.977	0.972	0.925	0.945	0.953
	MCL-GAN	10	0.976	0.975	0.964	0.914	0.940	0.938
LSGAN (Mao et al., 2017)	Base	1	0.977	0.957	0.928	0.866	0.923	0.943
	GMAN (Durugkar et al., 2017)	10	0.966	0.973	0.953	0.952	0.934	0.906
	MCL-GAN	10	0.983	0.980	0.963	0.911	0.950	0.952
Hinge (Lim & Ye, 2017)	Base	1	0.790	0.785	0.936	0.853	0.905	0.883
	MCL-GAN	5	0.957	0.965	0.959	0.916	0.914	0.925
	MCL-GAN	10	0.978	0.968	0.949	0.885	0.928	0.931

Table 2: FID scores on CIFAR-10 with the DCGAN architecture.

Model	# Disc. (m)	# Gen.	FID ↓
DCGAN Radford et al. (2016)	1	1	37.7
GMAN Durugkar et al. (2017)	10	1	37.11
Albuquerque *et al.* Albuquerque et al. (2019)	10	1	30.26
MGAN Hoang et al. (2018)	1	10	26.7
MSGAN Mao et al. (2019) (conditional)	1	1	28.73
MCL-GAN	10	1	26.87

validation set following (Yu et al., 2020). With the DCGAN architecture, we apply our method on three different GAN loss functions: the vanilla GAN (Goodfellow et al., 2014), LSGAN (Mao et al., 2017) and Hinge loss (Lim & Ye, 2017). Appendix I describes more details of our setting.

5.2.1 Quantitative results

We present the quantitative performance of MCL-GAN with the DCGAN and StyleGAN2 backbones using Precision Recall Distribution (PRD) (Sajjadi et al., 2018) and Frèchet Inception Distance (FID) (Heusel et al., 2017). More details about the evaluation metrics is provided in Appendix J.

DCGAN backbone Table 1 summarizes the precision and recall scores of our methods compared to the baseline models with different GAN objectives. MCL-GAN achieves outstanding performance in terms of both recall and precision compared to the baseline and GMAN on MNIST and CelebA. For Fashion-MNIST, we observe the different property of our method from GMAN while both methods surpass their baseline models; MCL-GAN focuses on improving the mode coverage (diversity) and GMAN cares about the image quality more than the diversity. Among many combinations of the number of discriminators (m) and experts (k) for our method, we discuss the results when $m = 5, 10$ and $k = 1$ for the moment and leave the thorough analysis on the hyperparameters in Appendix F.

Table 2 compares the FID scores on CIFAR-10 with other GAN models. MCL-GAN outperforms DCGAN, GMAN and Albuquerque et al. (2019) by large margins while it is as competitive as MGAN. This is encouraging because MGAN relies on multiple generators, 10 in this case. MSGAN results are obtained given class labels. This result implies that MCL-GAN is effective to maintain the multi-modality in the underlying distribution with relatively small memory footprint and without extra supervision.

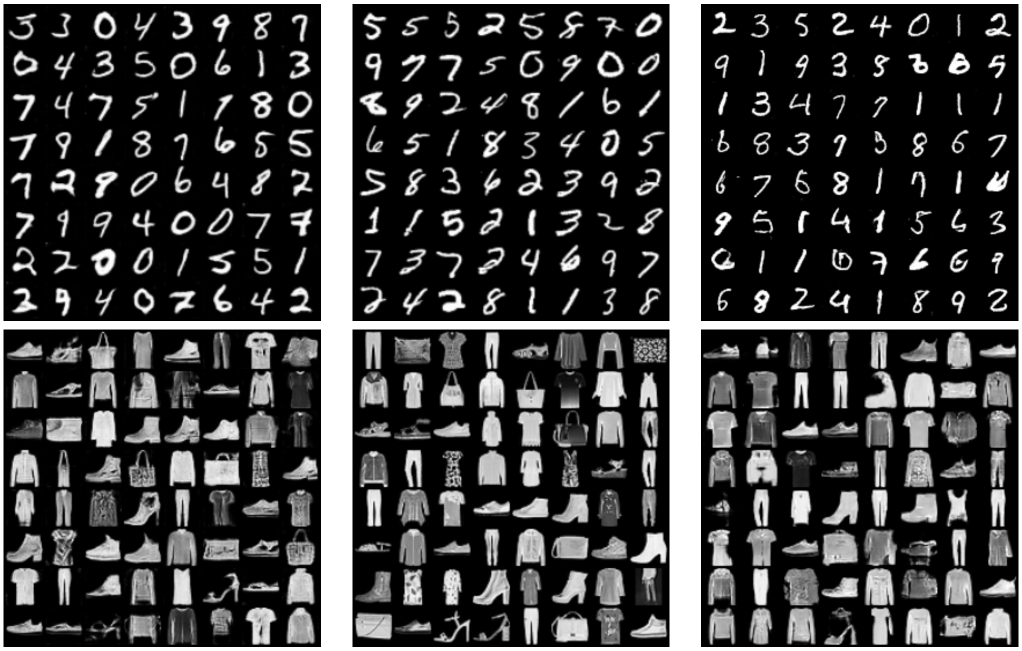
StyleGAN2 backbone Table 3 presents that MCL-GAN is also effective in the state-of-the-art backbone model, StyleGAN2, and outperforms not only StyleGAN2 but also Inclusive GAN (Yu et al., 2020) in terms of all metrics. For CelebA30K, we evaluates the performances on both train and validation sets. Note that Inclusive GAN uses the sample-wise reconstruction loss by regarding each image as a mode, which appears to improve recall. However, note that this goal is different from the objective of the standard GAN, estimating the underlying distribution. Also, the model may suffer from sampling bias and scalability issue.

5.2.2 Qualitative results

We investigate the quality of images generated by MCL-GAN and compare its performance with GMAN (Durugkar et al., 2017), which is an existing approach based on multiple discriminators.

Table 3: FID, precision and recall scores on CIFAR-10 and CelebA datasets with the StyleGAN2 architecture, where 10 and 5 discriminators are adopted, respectively, while $k = 1$. The asterisk ($*$) means that results are copied from (Yu et al., 2020) except for our method.

	CIFAR-10			CelebA30K*					
Method	FID↓	Rec.↑	Prec.↑	FID↓		Rec.↑		Prec.↑	
	-	-	-	Train	Val	Train	Val	Train	Val
StyleGAN2 (Karras et al., 2020)	9.06	0.979	0.984	9.37	9.49	0.730	0.741	0.855	0.844
Inclusive GAN (Yu et al., 2020)	-	-	-	11.56	11.28	0.849	0.848	0.927	0.941
MCL-GAN	7.13	0.985	0.989	8.41	8.61	0.988	0.990	0.985	0.983



(a) MCL-GAN (b) Real (c) GMAN

Figure 3: Qualitative comparison between MCL-GAN and GMAN on MNIST (top) and Fashion-MNIST (bottom). MCL-GAN generates more semantically faithful and diverse images than GMAN.

Figure 3 illustrates clear difference between MCL-GAN and GMAN on MNIST and Fashion-MNIST. For MNIST, the generated images by GMAN is sometimes hard to recognize or too thin and crisp compared to the real examples. The images for Fashion-MNIST are lacking in diversity; the types of generated bags and shoes are rather simple. On the other hand, MCL-GAN generates the images that are faithful to the true distribution in semantics and diversity and are indistinguishable from real images. More qualitative results are available in Appendix C.1.

5.3 Conditioned image synthesis

We apply the MCL-GAN to image-to-image translation and text-to-image synthesis tasks, which require more complex architectures to generate high-resolution images. In this experiment, the mode-seeking regularizer introduced in MSGAN (Mao et al., 2019) has been applied to alleviate the mode collapse issue in conditional GANs. Then, we observe whether the mode seeking technique and the use of multiple discriminators create synergy, using FID, NDB/JSD (Richardson & Weiss, 2018), and LPIPS (Zhang et al., 2018) following (Mao et al., 2019).

Image-to-image translation We choose DRIT (Lee et al., 2018; 2020) as our baseline, which is an unpaired image-to-image translation technique based on the cycle consistency. We employ MCL-GAN with $m = 3$ and $k = 1$ in each discriminator for distinguishing the real and the translated images. As shown in Table 4, MCL-GAN significantly improves the diversity measure, LPIPS,

Table 4: Quantitative results on Yosemitee (Summer⇌Winter) and Cat⇌Dog dataset. The best results are obtained when MCL component is added in most cases. The asterisk (∗) means that results are copied from (Mao et al., 2019).

Dataset	Metric	DRIT*	+MS (DRIT++)*	+MCL	+MCL+MS
Winter → Summer	FID ↓	47.37 ± 3.25	46.23 ± 2.45	49.41 ± 1.29	**41.94 ± 1.43**
	NDB ↓	30.60 ± 2.97	27.80 ± 3.03	**23.40 ± 1.52**	24.20 ± 3.27
	JSD ↓	0.049 ± 0.009	0.038 ± 0.004	0.033 ± 0.002	**0.030 ± 0.005**
	LPIPS ↑	0.097 ± 0.000	0.118 ± 0.001	0.153 ± 0.001	**0.248 ± 0.001**
Dog → Cat	FID ↓	62.85 ± 0.21	29.57 ± 0.23	**20.61 ± 0.05**	27.16 ± 0.20
	NDB ↓	41.00 ± 0.71	31.00 ± 0.71	**16.40 ± 0.89**	20.20 ± 1.48
	JSD ↓	0.272 ± 0.002	0.068 ± 0.001	**0.024 ± 0.001**	0.031 ± 0.001
	LPIPS ↑	0.102 ± 0.001	0.214 ± 0.001	0.429 ± 0.001	**0.482 ± 0.000**

Table 5: Quantitative results on CUB-200-2011. We obtained improved results consistently by adding the proposed MCL component. The asterisk (∗) means that results are copied from (Mao et al., 2019).

	StackGAN++*	+MS*	+MCL	+MCL+MS
FID ↓	25.99 ± 4.26	25.53 ± 1.83	**22.91 ± 0.80**	25.44 ± 0.41
NDB ↓	38.20 ± 2.39	30.60 ± 2.51	28.80 ± 3.63	**23.20 ± 3.03**
JSD ↓	0.092 ± 0.005	0.073 ± 0.003	0.079 ± 0.004	**0.053 ± 0.002**
LPIPS ↑	0.362 ± 0.004	0.373 ± 0.007	**0.629 ± 0.001**	0.624 ± 0.002

while achieving high-fidelity data generation performance in terms of other metrics. In particular, our approach works better on more challenging task, cat⇌dog, due to object shape changes across domains. Table 6 presents the translation results in the opposite directions on the two datasets.

Text-to-image synthesis This experiment is based on StackGAN++ (Zhang et al., 2017) trained on CUB-200-2011 (Wah et al., 2011) with a mode-seeking regularizer. StackGAN++ has a hierarchical structure that each set of a discriminator and a generator is responsible for a certain resolution. We adopt its 3-stage version and trains an MCL-GAN with $m = 3$ and $k = 1$ only at the last stage, which handles images with size 256×256. Table 5 illustrates that the integration of MCL improves performance consistently, especially in terms of the diversity measure, LPIPS.

6 Discussion

MCL-GAN is a model-agnostic ensemble algorithm with multiple discriminators. Our experiments imply that the specialized discriminators on the well-clustered subsets are beneficial compared to independently trained ones on the whole dataset or its random subsets. Although MCL-GAN does not rely on class labels for discriminator specialization, its performance is as competitive as the discriminator assignment based on the class labels (see Appendix E). The proposed approach runs efficiently because it is free from any time-consuming clustering procedure for sample assignment.

One drawback is that our method carries additional hyperparameters including the weights for several loss terms and the number of discriminators, and one might question about the robustness of MCL-GAN with respect to the variations of the hyperparameters. From our analysis on the hyperparameter setting, presented in Appendix F, the performance of the proposed method improves significantly by the expert training and the balanced assignment of discriminators while the rest of the loss terms make stable contributions over a wide range of their weights. Also, since MCL-GAN adjusts the number of active discriminators that participate in learning as experts, its performance is robust to the number of discriminators.

7 Conclusion

We presented a generative adversarial network framework with multiple discriminators, where each discriminator behaves as an expert classifier and covers a separate mode in the underlying distribution. This idea is implemented by incorporating the concept of multiple choice learning. The combination of generative adversarial network and multiple choice learning turns out to be effective to alleviate the mode collapse problem. Also, the integration of the sparsity loss encourages our model to identify the proper number of discriminators and estimate a desirable distribution. We demonstrated the effectiveness of the proposed algorithm on various GAN models and datasets.

Reproducibility statement We provide implementation and evaluation details in Appendix I and J to facilitate reproduction of the results presented in Section 5. The source code is available in the supplementary material. We will release the code.

Ethics statement Deep generative models have some potentials to be used for adverse or abusive applications. Although our work involves unconditional image generations based on face datasets, this is rather a generic framework based on GANs to mitigate the mode collapse and dropping problems hampering sample diversity. Our algorithm is not directly related to particular applications with ethical issues, and we believe that the proposed approach can alleviate the bias and fairness issues by identifying the minority groups in a dataset effectively.

REFERENCES

Isabela Albuquerque, Joao Monteiro, Thang Doan, Breandan Considine, Tiago Falk, and Ioannis Mitliagkas. Multi-objective training of generative adversarial networks with multiple discriminators. In *ICML*, pp. 202–211, 2019.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. volume 70 of *PMLR*, pp. 214–223, International Convention Centre, Sydney, Australia, 06–11 Aug 2017.

Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *ICCV*, pp. 1511–1520, 2017.

Thang Doan, Joao Monteiro, Isabela Albuquerque, Bogdan Mazoure, Audrey Durand, Joelle Pineau, and R Devon Hjelm. On-line adaptative curriculum learning for gans. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3470–3477, 2019.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. 2017.

Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *ICLR*, 2017.

Ishan Durugkar, Ian Gemp, and Sridhar Mahadevan. Generative multi-adversarial networks. 2017.

Michael Firman, Neill DF Campbell, Lourdes Agapito, and Gabriel J Brostow. Diversenet: When one right answer is not enough. In *CVPR*, pp. 5598–5607, 2018.

Arnab Ghosh, Viveka Kulharia, Vinay P Namboodiri, Philip HS Torr, and Puneet K Dokania. Multi-agent diverse generative adversarial networks. In *CVPR*, pp. 8513–8521, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pp. 2672–2680, 2014.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NeurIPS*, pp. 5767–5777, 2017.

Abner Guzman-Rivera, Dhruv Batra, and Pushmeet Kohli. Multiple choice learning: Learning to produce multiple structured outputs. In *NeurIPS*, pp. 1799–1807, 2012.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. In *NeurIPS*, 2017.

Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Phung. Mgan: Training generative adversarial nets with multiple generators. In *ICLR*, 2018.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, pp. 8110–8119, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pp. 4681–4690, 2017.

Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018.

Hsin-Ying Lee, Hung-Yu Tseng, Qi Mao, Jia-Bin Huang, Yu-Ding Lu, Maneesh Kumar Singh, and Ming-Hsuan Yang. Drit++: Diverse image-to-image translation via disentangled representations. *IJCV*, pp. 1–16, 2020.

Kimin Lee, Changho Hwang, Kyoung Soo Park, and Jinwoo Shin. Confident multiple choice learning. In *ICML*, pp. 2014–2023, 2017.

Stefan Lee, Senthil Purushwalkam Shiva Prakash, Michael Cogswell, Viresh Ranjan, David Crandall, and Dhruv Batra. Stochastic multiple choice learning for training diverse deep ensembles. In *NeurIPS*, pp. 2119–2127, 2016.

Ke Li, Tianhao Zhang, and Jitendra Malik. Diverse image synthesis from semantic layouts via conditional imle. In *ICCV*, pp. 4220–4229, 2019.

Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. *NeurIPS*, 2018.

Shaohui Liu, Xiao Zhang, Jianqiao Wangni, and Jianbo Shi. Normalized diversification. In *CVPR*, pp. 10306–10315, 2019.

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

Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *CVPR*, pp. 1429–1437, 2019.

Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, pp. 2794–2802, 2017.

Jonghwan Mun, Kimin Lee, Jinwoo Shin, and Bohyung Han. Learning to specialize with knowledge distillation for visual question answering. In *NeurIPS*, pp. 8081–8091, 2018.

Behnam Neyshabur, Srinadh Bhojanapalli, and Ayan Chakrabarti. Stabilizing gan training with multiple random projections. *arXiv preprint arXiv:1705.07831*, 2017.

Tu Nguyen, Trung Le, Hung Vu, and Dinh Phung. Dual discriminator generative adversarial nets. In *NeurIPS*, pp. 2670–2680, 2017.

David Keetae Park, Seungjoo Yoo, Hyojin Bahng, Jaegul Choo, and Noseong Park. Megan: Mixture of experts of generative adversarial networks for multimodal image generation. In *IJCAI*, pp. 878–884, 2018.

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

Eitan Richardson and Yair Weiss. On gans and gmms. *arXiv preprint arXiv:1805.12462*, 2018.

Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lučić, Olivier Bousquet, and Sylvain Gelly. Assessing Generative Models via Precision and Recall. In *NeurIPS*, pp. 5228–5237, 2018.

Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *NeurIPS*, pp. 3308–3318, 2017.

Kai Tian, Yi Xu, Shuigeng Zhou, and Jihong Guan. Versatile multiple choice learning and its application to vision computing. In *CVPR*, pp. 6349–6357, 2019.

Ilya O Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann Simon-Gabriel, and Bernhard Schölkopf. Adagan: Boosting generative models. In *NeurIPS*, pp. 5424–5433, 2017.

Quan Hoang Trung Le, Hung Vu, Tu Dinh Nguyen, Hung Bui, and Dinh Phung. Learning generative adversarial networks from multiple data sources. In *IJCAI*, pp. 2823–2829, 2019.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Dingdong Yang, Seunghoon Hong, Yunseok Jang, Tianchen Zhao, and Honglak Lee. Diversity-sensitive conditional generative adversarial networks. In *ICLR*, 2018.

Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *CVPR*, pp. 5485–5493, 2017.

Ning Yu, Ke Li, Peng Zhou, Jitendra Malik, Larry Davis, and Mario Fritz. Inclusive gan: Improving data and minority coverage in generative models. In *European Conference on Computer Vision*, pp. 377–393. Springer, 2020.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, pp. 5907–5915, 2017.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

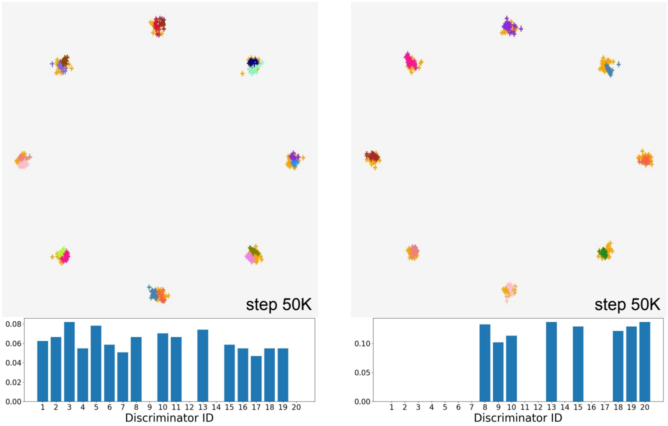
Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial networks. In *ICLR*, 2017.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017a.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pp. 2223–2232, 2017b.

A Effect of L_1 loss on synthetic data

To examine the behaviour of L_1 loss, we run the experiment with 20 discriminators which exceeds the actual number of modes of 8 Gaussians dataset. Figure 4 shows each expert discriminator per generated sample by different colors. Training with $m = 20$ without L_1 loss, 16 discrimantors are utilized to cluster the true distribution. Since two discrimantors are assigned per each mode, the diversity within the mode is improved. By adding a small L_1 loss, we discover only 8 discrimantors are effectively used in training, one for each mode. These results show that the L_1 regularization helps identify the proper number of discriminators to generate high-fidelity data efficiently.



(a) $\gamma = 0$ (b) $\gamma = 0.00001$

Figure 4: Effect of L_1 loss weight (γ). Each random sample is colored by its expert discriminator. True data are in orange. Bar graphs demonstrate the update statistics of individual discriminators.

B Quantitative results

We include the all image-to-image translation results on Yosemitee (Summer⇌Winter) and cat⇌dog dataset in Table 6 in addition to Table 4 of the main paper. MCL-GAN improves the diversity measure (LPIPS) for all cases while achieving better or competitive quality of images.

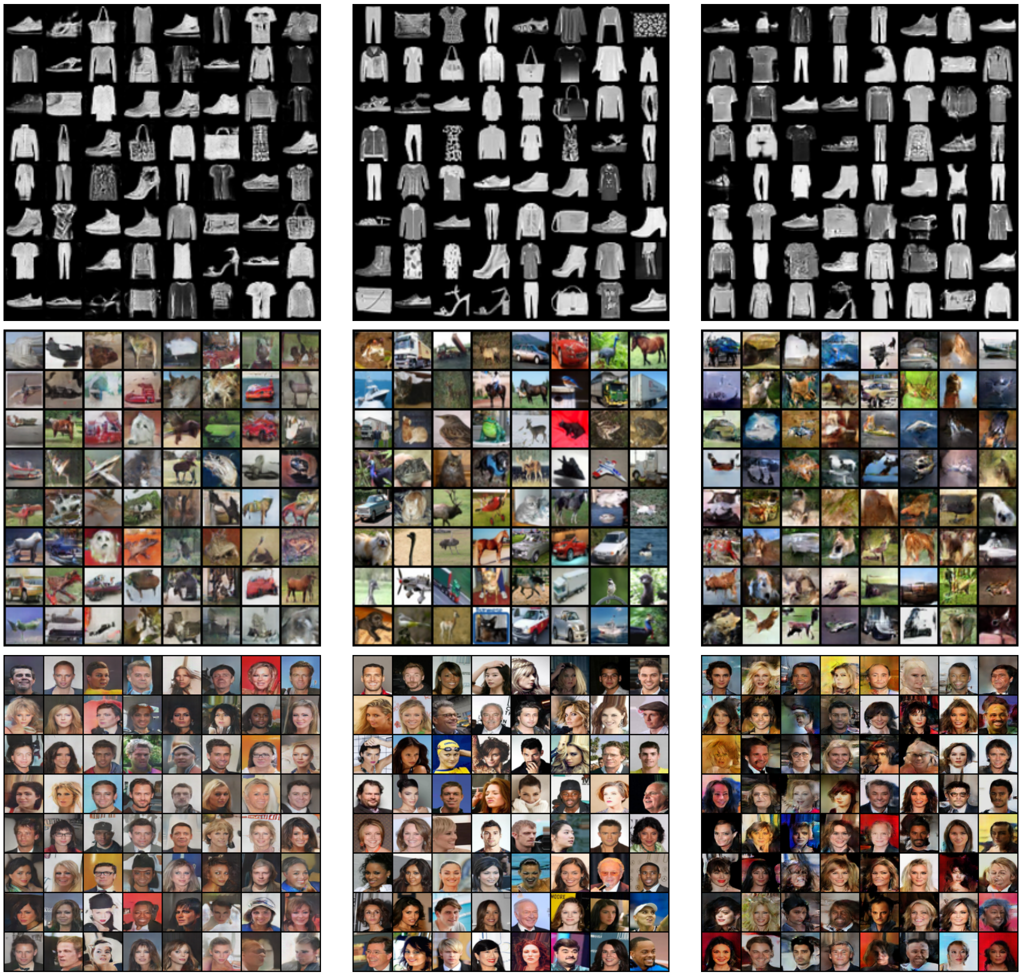
Table 6: Quantitative results on Yosemitee (Summer⇌Winter) and Cat⇌Dog dataset. The best results are obtained when MCL component is added in most cases. The asterisk ($*$) means that results are copied from (Mao et al., 2019).

Dataset	Metric	DRIT (Lee et al., 2018)*	+MS (DRIT++)*	+MCL	+MCL+MS
Summer → Winter	FID ↓	57.24 ± 2.03	51.85 ± 1.16	53.77 ± 1.36	**49.74 ± 2.74**
	NDB ↓	25.60 ± 1.14	**22.80 ± 2.96**	25.40 ± 1.14	30.00 ± 2.55
	JSD ↓	0.066 ± 0.005	0.046 ± 0.006	**0.036 ± 0.004**	0.044 ± 0.005
	LPIPS ↑	0.115 ± 0.000	0.147 ± 0.001	0.199 ± 0.002	**0.263 ± 0.003**
Winter → Summer	FID ↓	47.37 ± 3.25	46.23 ± 2.45	49.41 ± 1.29	**41.94 ± 1.43**
	NDB ↓	30.60 ± 2.97	27.80 ± 3.03	**23.40 ± 1.52**	24.20 ± 3.27
	JSD ↓	0.049 ± 0.009	0.038 ± 0.004	0.033 ± 0.002	**0.030 ± 0.005**
	LPIPS ↑	0.097 ± 0.000	0.118 ± 0.001	0.153 ± 0.001	**0.248 ± 0.001**
Cat → Dog	FID ↓	22.74 ± 0.28	16.02 ± 0.30	20.64 ± 0.13	**15.36 ± 0.16**
	NDB ↓	42.00 ± 2.12	27.20 ± 0.84	29.80 ± 1.10	**22.20 ± 2.77**
	JSD ↓	0.127 ± 0.003	0.084 ± 0.002	0.048 ± 0.002	**0.031 ± 0.002**
	LPIPS ↑	0.245 ± 0.002	0.280 ± 0.002	0.511 ± 0.000	**0.553 ± 0.000**
Dog → Cat	FID ↓	62.85 ± 0.21	29.57 ± 0.23	**20.61 ± 0.05**	27.16 ± 0.20
	NDB ↓	41.00 ± 0.71	31.00 ± 0.71	**16.40 ± 0.89**	20.20 ± 1.48
	JSD ↓	0.272 ± 0.002	0.068 ± 0.001	**0.024 ± 0.001**	0.031 ± 0.001
	LPIPS ↑	0.102 ± 0.001	0.214 ± 0.001	0.429 ± 0.001	**0.482 ± 0.000**

C Qualitative results

C.1 Unconditional GAN on image datasets

Figure 5 compares the results of generated examples by MCL-GAN and GMAN together with real data on Fashion-MNIST (Xiao et al., 2017), CIFAR-10 (Krizhevsky et al., 2009) and CelebA (Liu et al., 2015). The samples are drawn randomly rather than cherry-picked. For CIFAR-10, MCL-GAN generate relatively clear images and some of them are recognizable as vehicles or animals (see Figure 6) whereas most images obtained from GMAN look incomplete and noisy. For CelebA, GMAN produces high quality images, but we discover more distorted and unnatural images than MCL-GAN. Some selected examples from MCL-GAN with DCGAN backbone on Fashion-MNIST, CIFAR-10, and CelebA are displayed in Figure 6. Figure7 shows random samples generated by MCL-GAN with StyleGAN2 backbone on CIFAR-10 and CelebA30K.

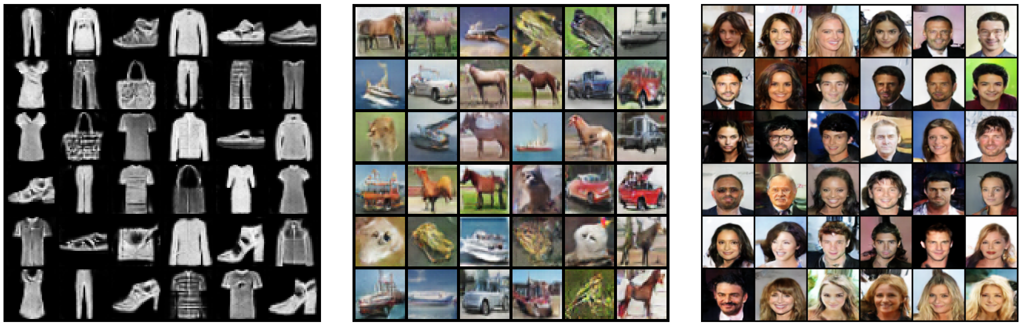


(a) MCL-GAN (b) Real (c) GMAN

Figure 5: Qualitative comparison between MCL-GAN and GMAN on Fashion-MNIST (top), CIFAR-10 (middle) and CelebA (bottom). MCL-GAN generates more realistic images with less failure cases than GMAN.

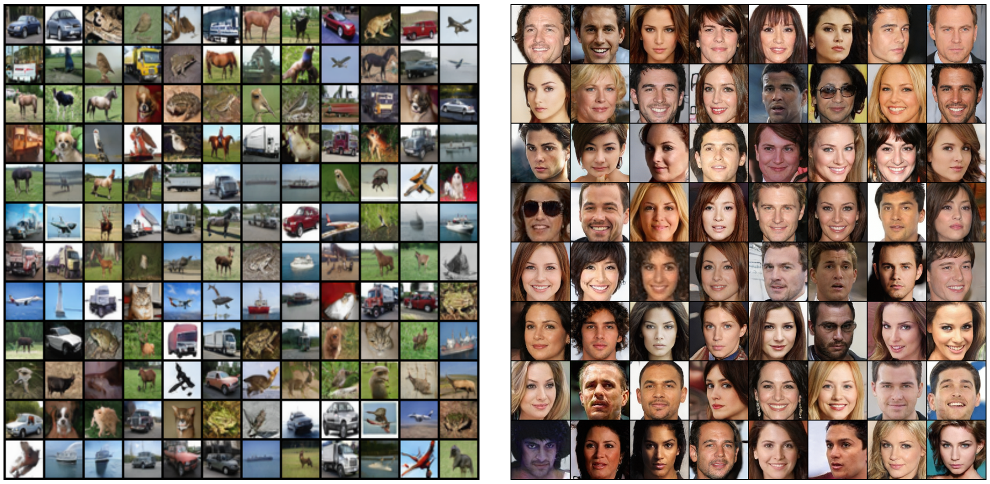
C.2 Conditioned image synthesis

Figure 11 and 12 qualitatively compare the diversity of the generated images between the baselines and MCL-GANs. For all methods including the baselines, mode-seeking regularization (Mao et al.,



(a) Fashion-MNIST (b) CIFAR-10 (c) CelebA

Figure 6: Selected samples generated by MCL-GAN with DCGAN architecture.



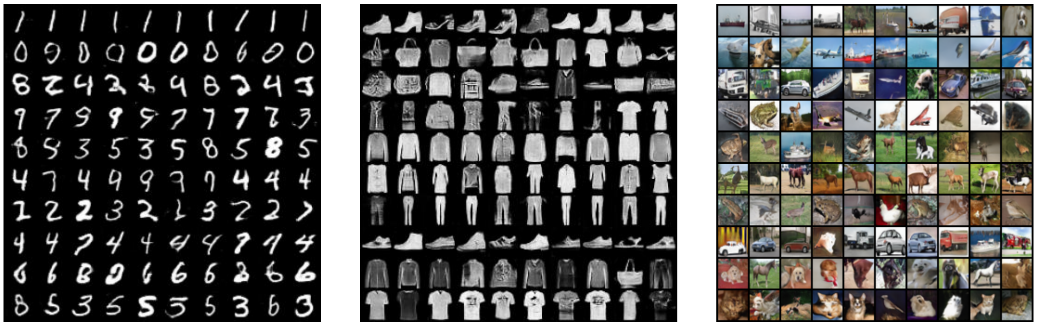
(a) CIFAR-10 (b) CelebA30K

Figure 7: Random samples generated by MCL-GAN with StyleGAN2 architecture. For generation, truncation $\psi = 0.8$ is applied.

2019) is applied. As shown in Figure 11(a), images generated by MCL-GAN have more variations in edges and expressions of dogs. Regarding Yosemitee results (Figure 11(b)) which the shapes of the contents are fixed, colors are more diverse and vivid in MCL-GAN results. For Figure 12, we fix the text code for each text description to remove the diversity effect of text embedding and produce images with the same set of latent vectors. MCL-GAN produces more diverse bird images, in terms of shape, orientation and size with high quality. We present more qualitative results of MCL-GAN in Figure 13 and 14.

D SPECIALIZATION OF EACH DISCRIMANTOR

Figure 8 qualitatively presents how successfully the discriminators in MCL-GAN are specialized to the subsets of the whole datasets. We learn the model with 10 discriminators, and illustrate the generated images. Note that the panel corresponding to each dataset consists of 10×10 images and the images in the same row belong to the same discriminators. We can observe semantic consistency of images within the same row in MNIST and Fashion-MNIST clearly. The images in the same row of CIFAR-10 also have some similarities although the signal is not as strong as the other two datasets. We believe that this is partly due to the inherent characteristics of the dataset that are more difficult to recognize.



(a) MNIST (DCGAN) (b) Fashion-MNIST (DCGAN) (c) CIFAR-10 (StyleGAN2)

Figure 8: Cluster by discrimantors. Each row represents the subcluster of each discriminator. Close similarities are discovered among the images if they are in the same row.

E Clustering via MCL vs. ground-truth label

We additionally run the same multi-discriminator framework by clustering using ground-truth labels instead of MCL on real dataset. This is to verify the effectiveness of the specialized discriminators on our learning framework and check if the clustering by MCL is sufficiently reliable compared to the results by true labels. As presented in Table 7, the performance of MCL is as competitive as the method based on true labels in terms of both metrics.

Table 7: Recall and precision scores given by different clustering methods: MCL vs. ground-truth label. The model 'Label' assigns an expert discriminator of each real sample by the ground-truth label under our multi-discriminator framework.

			MNIST		Fashion-MNIST	
Model	m	k	Recall	Precision	Recall	Precision
DCGAN	1	1	0.891	0.789	0.927	0.903
MCL-GAN	5	1	0.983	0.977	0.977	0.929
MCL-GAN	10	1	0.976	0.973	0.967	0.916
Label	10	1	0.978	0.966	0.969	0.935

F Analysis on hyperparameters

F.1 Non-expert loss weight

Table 8 shows the effect of non-expert training regularization when training 10 discriminators with the standard GAN loss on MNIST. The performance increase is mostly driven by expert training. The best score is obtained with $\alpha = 0.01$ while all positive α improves the precision scores, which supports the effect of the lowered confidence of non-expert discriminators.

Table 8: Effect of non-expert loss weight (α) when $m = 10$ and $k = 1$ on MNIST.

α	0	0.01	0.1	0.2	0.5	0.75	1
Recall	0.983	0.984	0.978	0.982	0.983	0.976	0.976
Precision	0.951	0.977	0.965	0.968	0.974	0.972	0.971

F.2 Balance loss weights

We denote the balance loss weights of discriminator and generator by β_d and β_g, respectively. We conduct the ablation studies on several (β_d, β_g) combinations for 10 discriminators with Hinge loss on MNIST. In Table 9, we observe that β_d plays an important role in boosting the performance of GAN. This is mainly because β_d is responsible for distributing the chances of being an expert to multiple discriminators; Only a few discriminators are utilized in training if β_d is zero or too small.

β_g has a relatively smaller effect on the performance than β_d. However, it helps improve recall scores without sacrificing precision as the best score is obtained when $(\beta_d, \beta_g) = (0.5, 10)$. Note that the performance does not change drastically for all cases where $\beta_d > 0$ and surpasses a single discriminator GAN with a large margin.

Table 9: Effect of balance loss weights (β_d and β_g) when $m = 10$ and $k = 1$ on MNIST.

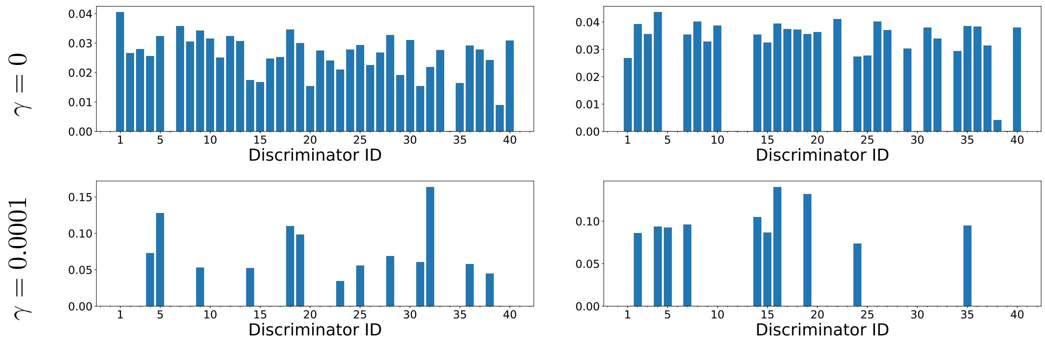
β_d	β_g	Recall	Precision
vanilla		0.803	0.765
0	0	0.926	0.856
0.2	0	0.973	0.966
0.5	0	0.971	0.970
0	5	0.931	0.894
0.2	5	0.978	0.963
0.5	5	0.977	0.967
0	10	0.949	0.883
0.2	10	0.978	0.966
0.5	10	0.981	0.972

F.3 Number of discrimantors and L_1 loss weight

We conduct the experiment under a various number of discriminators and illustrate the results in Table 10. It turns out that the performance of the proposed method is fairly robust to the number of discriminators quantitatively and adding the L_1 loss does not incur noticeable differences in terms of precision/recall measure. However, interestingly, the L_1 loss plays a crucial role in finding modes in the underlying distribution. Figure 9 illustrates the impact of the L_1 loss on MNIST and Fashion-MNIST when we train the model with 40 discriminators. According to our results, only a fraction of the discriminators are specialized to data, and the number of active discriminators is fairly coherent to the number of classes in the dataset.

Table 10: Comparisons by number of discriminators (m).

	MNIST		Fashion-MNIST	
m	Recall	Precision	Recall	Precision
1	0.883	0.795	0.928	0.904
5	0.979	0.976	0.974	0.929
10	0.974	0.972	0.965	0.934
20	0.972	0.958	0.958	0.922
40	0.977	0.970	0.974	0.938
20 (+L_1)	0.967	0.964	0.967	0.939
40 (+L_1)	0.973	0.960	0.966	0.914



(a) MNIST

(b) Fashion-MNIST

Figure 9: Effect of L_1 loss weight (γ). The update statistics of individual discriminators when 40 discriminators are used for training on MNIST and Fashion-MNIST datasets.

F.4 NUMBER OF EXPERTS PER EXAMPLE

We evaluate our model on the various numbers of experts k for $m = 5$ and 10, and present the results on MNIST and CIFAR-10 in Table 11. The number of optimal k may be different in each dataset, however, choosing too many experts tend to drop the scores relevant to recall metric.

Figure 10 shows how the specialization characteristics of discriminators differ by the number of experts per sample, *i.e.*, $k \in \{1, 3, 5\}$, when there are 10 discriminators on MNIST and Fashion-MNIST. As k increases, the models get less specialized by sharing more data each other so the subclusters become less distinctive.

Table 11: Comparisons by number of experts per sample (k).

		MNIST		CIFAR-10	
m	k	Recall	Precision	Recall	Precision
5	1	0.983	0.975	0.903	0.942
5	3	0.983	0.981	0.896	0.948
10	1	0.973	0.973	0.902	0.937
10	3	0.973	0.969	0.913	0.946
10	5	0.975	0.964	0.917	0.948
10	7	0.960	0.927	0.912	0.951

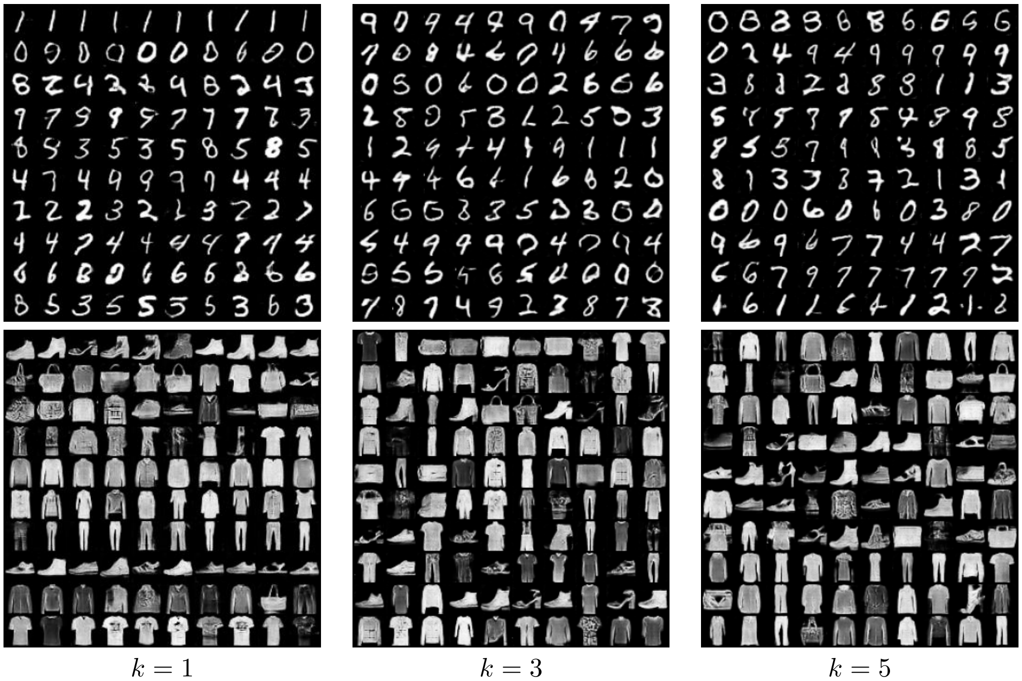


Figure 10: Specialization results for $k \in \{1, 3, 5\}$ with 10 discriminators on MNIST and Fashion-MNIST. The images in each row correspond to the same discriminators.

G STABILITY TO HYPERPARAMETER CHANGE

We conduct experiments on MNIST with $m = 10, 20, 40$ using L_1 regularization multiple times and observe that the accuracies (precision and recall) are very stable regardless of the number of discriminators for expert training as in Table 12. Note that we performed each experiment 4 times with random initialization.

Table 12: Stability of model performances and the number of active discriminators when $m = 10, 20, 40$ and $\gamma = 0.0002$ on MNIST.

m	Recall	Precision	# active discriminators
10 (+L_1)	0.965 ± 0.004	0.965 ± 0.006	8.3 ± 1.0
20 (+L_1)	0.966 ± 0.004	0.964 ± 0.002	12.0 ± 1.2
40 (+L_1)	0.968 ± 0.005	0.963 ± 0.009	10.0 ± 1.4

H Computational overheads

Table 13 compares the training time per iteration and memory usage with the DCGAN baselines when training on CelebA dataset (64×64 sized). While independent training of multiple discriminators, *e.g.*, GMAN (Durugkar et al., 2017), requires more than four times the resources and time, additional overheads are marginal for MCL-GAN due to feature sharing. We used a machine with a Titan Xp GPU for the measurement.

Table 13: Comparisons of computational overheads on CelebA.

	DCGAN ($m = 1$)	MCL-GAN ($m = 10$)	GMAN ($m = 10$)
Time (s/iteration)	0.4412	0.4584	2.6369
Memory (MB)	2443	2991	11857

I Implementation details

I.1 Synthetic data

We reuse the experimental design and implementation[1] following (Gulrajani et al., 2017).

I.2 Unconditional GAN on image datasets

DCGAN backbone We mostly follow the training convention proposed in DCGAN (Radford et al., 2016). We use Adam optimizer (Kingma & Ba, 2015) with $\beta = (0.5, 0.999)$ and set 64 and 128 as size of the mini-batch for real data and latent vectors, respectively. We use the same learning rate and temperature in balance loss for all networks, *i.e.*, $lr = 0.0001, \tau = 0.1$ for LSGAN experiments and $lr = 0.0002, \tau = 1.0$ for the others. The weights for balance loss of discrimantors, β_d, is chosen in the range $[0.05, 1.0]$ and we choose the best performance. For the weights for balance loss of generator, $\beta_g = 0$ produces fairly good results on all cases while positive β_g gives particularly significant improvement in some LSGAN and Hinge loss experiments. We choose $\beta_g \in \{1.0, 2.0\}$ for LSGAN experiments on all datasets and $\beta_g \in \{5.0, 10.0\}$ for Hinge loss experiments on MNIST. For implementing standard GAN loss, we use the modified minimax objective for the generator, *i.e.*, $\min -\mathbb{E}_{z\sim p_z} \log D(G(z))$.

StyleGAN2 backbone We adopt the configuration E architecture among the StyleGAN2 variations and use default hyperparameters for training using the official implementation[2] without applying data augmentation option. We set the batch size at 64 and 16 for CIFAR-10 and CelebA30K, respectively.

GMAN settings We used the official implementation[3] of GMAN. Among its varients, we use three versions that use the arithmetic mean of softmax, *i.e.*, GMAN-1, GMAN-0 and GMAN*, and choose the best scores among them to report in Table 1 and 2. For differentiating discriminators, we apply different dropout rates in $[0.4, 0.6]$ and split of mini-batches for the input of discriminators while adopting the same architectures as DCGAN.

[1] https://github.com/caogang/wgan-gp

[2] https://github.com/NVlabs/stylegan2-ada-pytorch

[3] https://github.com/iDurugkar/GMAN

I.3 CONDITIONED IMAGE SYNTHESIS

We apply the MCL components to the official codes of DRIT++[4], StackGAN++[5] and MSGAN[6] and use the default settings of their original implementations.

J EVALUATION DETAILS

J.1 UNCONDITIONAL GAN ON IMAGE DATASETS

Evaluation metrics We measure precision/recall based on Precision Recall Distribution (PRD) (Sajjadi et al., 2018). We adopt F_8 and $F_{1/8}$ scores from the PRD curve as a recall and precision of each model, respectively. We use the official implementations of PRD[7] and FID[8] for the measurement.

DCGAN backbone We run the the experiment on MNIST (LeCun & Cortes, 2010), Fashion-MNIST (Xiao et al., 2017), CIFAR-10 (Krizhevsky et al., 2009) and CelebA (Liu et al., 2015) until 40, 50, 150 and 30 epochs, respectively. We generate 60K random examples for MNIST and Fashion-MNIST and 50K random samples for the other datasets, and then compare them with the reference datasets with the same number of examples.

StyleGAN2 backbone We run the the experiment on CIFAR-10 (Krizhevsky et al., 2009) and CelebA30K (Liu et al., 2015) until 300 epochs and choose the best model in terms of FID. We generate 50K and 30K random examples for CIFAR-10 and CelebA30K, respectively, and then compare them with the whole train (/validation) set. We do not use the truncation trick when generating samples for quantitative evaluations.

J.2 CONDITIONED IMAGE SYNTHESIS

We measure FID, NDB/JSD[9] and LPIPS[10] using their official implementations. We follow all evaluation details in MSGAN (Mao et al., 2019) which is referenced for comparision. Note that NDB counts the number of statistically different bins based on the clusters made by k-means clustering while LPIPS measures the average feature distances of sample pairs. JSD is calculated based on the results (clusters) of NDB.

[4] https://github.com/HsinYingLee/DRIT

[5] https://github.com/hanzhanggit/StackGAN-v2

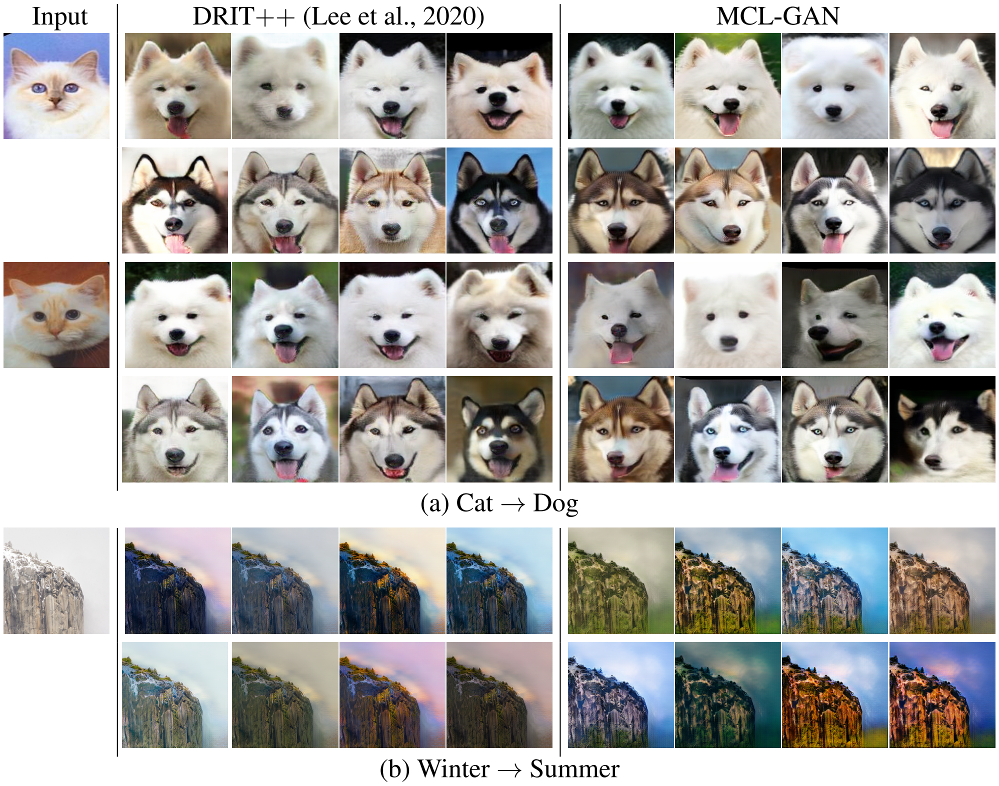
[6] https://github.com/HelenMao/MSGAN

[7] https://github.com/msmsajjadi/precision-recall-distributions

[8] https://github.com/bioinf-jku/TTUR

[9] https://github.com/eitanrich/gans-n-gmms

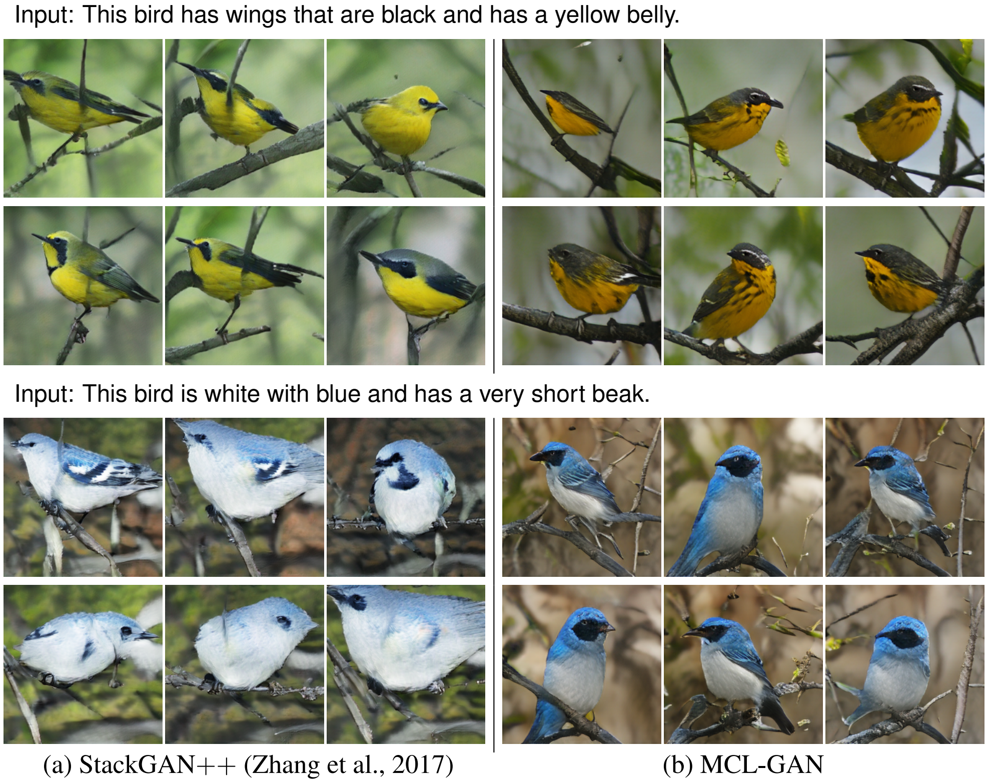
[10] https://github.com/richzhang/PerceptualSimilarity



(a) Cat → Dog

(b) Winter → Summer

Figure 11: Diversity comparison of image-to-image translation on Yosemitee (Summer⇌Winter) and Cat⇌Dog dataset.


Input: This bird has wings that are black and has a yellow belly.
Input: This bird is white with blue and has a very short beak.

(a) StackGAN++ (Zhang et al., 2017)

(b) MCL-GAN

Figure 12: Diversity comparison of text-to-image synthesis on CUB-200-2011.

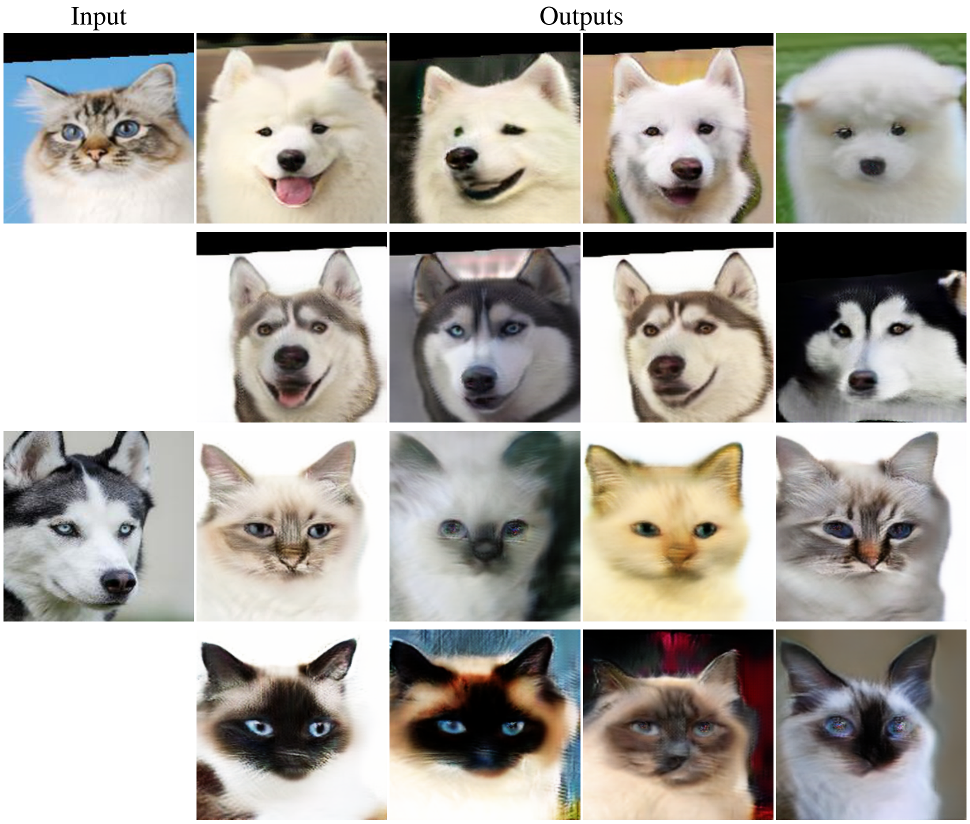
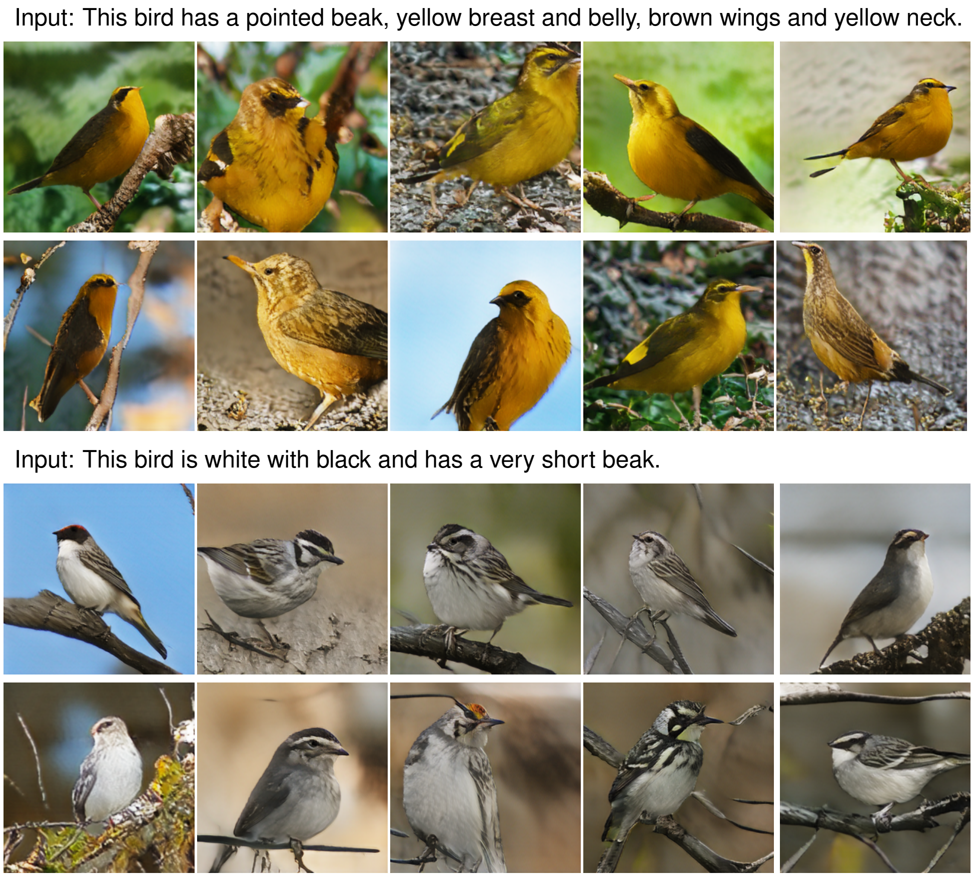


Figure 13: More image-to-image translation results by MCL-GAN on Cat$\rightleftharpoons$Dog.



Input: This bird has a pointed beak, yellow breast and belly, brown wings and yellow neck.

Input: This bird is white with black and has a very short beak.

Figure 14: More text-to-image synthesis results by MCL-GAN on CUB-200-2011.