Mining Multi-Label Samples from Single Positive Labels

Youngin Cho*[1] **Daejin Kim*[1,2]** **Mohammad Azam Khan[1]** **Jaegul Choo[1]**
[1]KAIST AI [2]NAVER WEBTOON AI
{choyi0521,kiddj,azamkhan,jchoo}@kaist.ac.kr

Abstract

Conditional generative adversarial networks (cGANs) have shown superior results in class-conditional generation tasks. To simultaneously control multiple conditions, cGANs require multi-label training datasets, where multiple labels can be assigned to each data instance. Nevertheless, the tremendous annotation cost limits the accessibility of multi-label datasets in real-world scenarios. Therefore, in this study we explore the practical setting called the *single positive setting*, where each data instance is annotated by only one positive label with no explicit negative labels. To generate multi-label data in the single positive setting, we propose a novel sampling approach called single-to-multi-label (S2M) sampling, based on the Markov chain Monte Carlo method. As a widely applicable "add-on" method, our proposed S2M sampling method enables existing unconditional and conditional GANs to draw high-quality multi-label data with a minimal annotation cost. Extensive experiments on real image datasets verify the effectiveness and correctness of our method, even when compared to a model trained with fully annotated datasets.

1 Introduction

Since being proposed by Goodfellow *et al.* [1], generative adversarial networks (GANs) have gained much attention due to their realistic output in a wide range of applications, *e.g.*, image synthesis [2, 3, 4], image translation [5, 6, 7, 8], and data augmentation [9, 10]. As an advanced task, generating images from a given condition has been achieved by conditional GANs (cGANs) and their variants [11, 12]. To reflect the nature of real data where each data instance can belong to multiple classes, multi-label datasets have been introduced in the applications of cGANs [8, 13]. In a multi-label dataset, each data instance can be specified with multiple attributes. For example, in a multi-label facial image dataset, each image is labeled for entire classes such as *Black-hair*, *Smile*, and *Male*. The label for each class is given as a binary value and indicates the presence or absence of the corresponding attribute. Despite its usefulness, as claimed in earlier studies [14, 15, 16], access to multi-label datasets is severely limited in practice due to the difficulty of annotating all attributes. Under these circumstances, a weakly annotated dataset is used as an alternative for cGANs to reduce the annotation cost.

In this paper, we introduce the *single positive setting* [16], originally proposed for classification tasks, to conditional generation. Each data instance has a label indicating only the presence of one attribute (*i.e.*, a single positive label) in this setting, and the presence of the rest of the attributes remains unknown. For instance, in a facial image dataset, all attributes are fully specified in a multi-label setting (*e.g.*, *Smiling black-haired man*) whereas only one attribute is specified in the single positive setting (*e.g.*, *Black-hair*). The single positive setting allows us not only to reduce the annotation cost, but also to model the intrinsic relationships among classes.

*Equal contribution

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

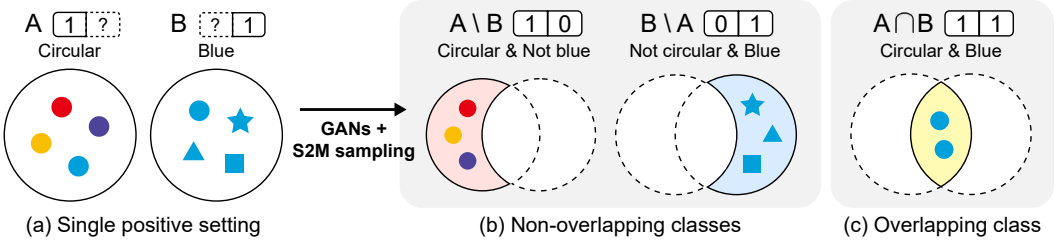


Figure 1: Illustration of joint classes where two attributes (circular, blue) are given. (a) Only one of the attributes is specified in the single positive setting. From classes A and B, our proposed S2M sampling method can draw samples from three joint classes: (b) two non-overlapping classes ($A \setminus B$ and $B \setminus A$) and (c) one overlapping class ($A \cap B$).

To generate multi-label data using only single positive labels, we consider two types of combinatorial classes from two classes, A and B, each corresponding to a single positive label: (1) *overlapping class*, where the data instances belong to both classes ($A \cap B$), and (2) *non-overlapping class*, where the data instances belong to only one of the classes ($A \setminus B$ or $B \setminus A$). Figure 1 shows examples of overlapping class and non-overlapping classes when two types of single positive labels are given. By extending this concept, we can consider combinatorial classes, where data instances belong to certain classes but not to the rest. We denote these classes as *joint classes*. By accessing a joint class, we can generate multi-label samples using only single positive labels. Ideally, we can represent all possible label combinations with at least one positive label.

Several attempts have been made to consider such a joint class in generation tasks. Specially designed generative models such as GenPU [17] and RumiGAN [18] were proposed to generate samples from one class while excluding another class. However, these studies deal with only two classes without considering the overlapping class. Recently, Kaneko *et al.* [19] proposed CP-GAN to capture between-class relationships in conditional generation. To generate images belonging to n classes, they equally assign $1/n$ as the class posterior for each selected class. This indicates that CP-GAN generates images that have an equal probability of being classified as each class. However, because an image in the real world that belongs to multiple classes does not have equal class posteriors, sample diversity is lost. Consequently, we focus on a sampling approach to precisely draw samples from complex distributions that are difficult to directly sample. In recent GANs studies, sampling approaches [20, 21] employed the rejection sampling or Markov chain Monte Carlo method to obtain realistic data. These approaches adopt the sampling process as the post-processing method of GANs and filter generated images using the pretrained classifiers.

In line with these studies, we propose a novel sampling framework called *single-to-multi-label (S2M) sampling* to correctly generate data from both overlapping and non-overlapping classes. We newly propose a tractable formulation to estimate the conditional density of joint classes. Concretely, we employ several classification networks to estimate the target conditional density and apply the Markov chain Monte Carlo method to pretrained unconditional and conditional GANs. In Figure 2, we depict the conceptual difference against the existing approaches (1st row) and provide the empirical results on a 1D Gaussian example (2nd row). As S2M sampling performs at the inference time of generation, it fully preserves the image quality and eliminates the need for changing the objective functions and architectures. We validate the effectiveness of our method compared to the existing models on diverse datasets such as MNIST, CIFAR-10 and CelebA. To the best of our knowledge, our approach is the first sampling framework that generates multi-label data from single positive labels. Our contributions can be summarized as follows:

- We introduce the single positive setting in the conditional generation task and provide the theoretical framework for estimating conditional densities of joint classes.
- We propose a novel sampling framework based on the Markov chain Monte Carlo method for generating multi-label data from single positive labels.
- Through extensive experiments, we show that our proposed S2M sampling method correctly draws multi-label data while fully preserving the quality of generated images.

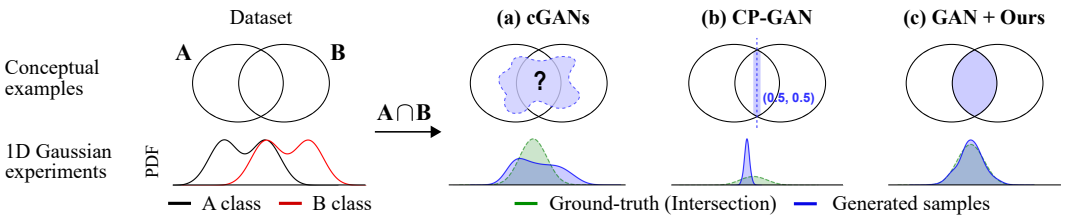


Figure 2: cGANs, CP-GAN and our method are compared in a class-overlapping case. 1D Gaussian examples consists of two classes of one-dimensional Gaussian mixtures with one common mode, and each method attempts to generate samples in the overlapping region. For cGANs and CP-GAN, an equal value of 0.5 is provided as labels for the two classes. (a) It is not clear how cGANs obtain samples of the class. (b) CP-GAN draws samples from the narrow region. (c) GAN with S2M sampling draws samples accurately without sacrificing diversity.

2 Related Work

Conditional GANs. The aim of conditional GANs (cGANs) [11] is to model complex distributions and control data generation by reflecting the label input. Various studies of cGANs have made significant advances in class-conditional image generation by introducing an auxiliary classifier [12, 22], modifying the architecture [23, 2], and applying metric learning [24]. In a weakly-supervised setting, GenPU [17] and RumiLSGAN [18] specify only two classes and draw samples that belong to one class but not the other. CP-GAN [19] learns to draw samples conditioned on the probability output of the classifier. Given that this model tends to draw samples on a limited region of the data space, it is challenging to ensure a variety of samples as shown in Figure 2. In contrast, we propose a sampling method that draws multi-label samples without sacrificing diversity.

Sampling in GANs. Sampling methods are used to improve the sample quality in GANs. Discriminator rejection sampling [20] uses the scheme of rejection sampling and takes samples close to real data by estimating the density ratio with the discriminator. In addition, Metropolis-Hastings GAN [21] adopts the Markov chain Monte Carlo (MCMC) method and calibrates the discriminator to improve the sample quality in a high-dimensional data space. Discriminator driven latent sampling [25] uses the MCMC method in the latent space of GANs to draw realistic samples efficiently. GOLD estimator [26] uses a sampling algorithm to improve the quality of images for class-conditional generation. While previous studies focus on improving the sample quality, our S2M sampling method aims to draw multi-label samples while also improving sample quality.

3 Methods

3.1 Problem Setting

Let $x \in X$ be a data point as a random variable and let $y_{1:n} \in \{0,1\}^n$ denote its corresponding multi-labels as binary random variables. Here, for every k, $y_k = 1$ indicates that x is contained in the k-th class while $y_k = 0$ indicates that x is not. We consider two index sets, an intersection index set I and a difference index set J, so that the pair (I, J) can be used as an index to indicate the class, where data points contained in all classes indicated by I but excluded from all classes indicated by J. Let $\mathcal{I}$ be a collection of all possible pairs of I and J, defined as

$$\mathcal{I} = \{(I, J) \in \mathcal{P}(N) \times \mathcal{P}(N) : I \neq \emptyset, I \cap J = \emptyset\}, \tag{1}$$

where $N = \{1, 2, ..., n\}$ is a finite index set of all classes and $\mathcal{P}(N)$ is the power set of N. That is, the intersection index set indicates at least one class and is distinct from the difference index set. Especially for $I \cup J = N$, the class indicated by (I, J) is called the *joint class*.

Let $p(x, y_1, y_2, ..., y_n)$ be the joint probability density function, and let $p_{data}(x, c)$ be the joint density of an observed data point $x \in X$ and a class label $c \in N$ such that $p_{data}(x|c) = p(x|y_c = 1)$. Given the class priors $p_{data}(c), \pi_c := p(y_c = 1)$ and samples drawn from the class-conditional density $p_{data}(x|c)$ for each $c = 1, 2, ..., n$, our goal is to draw samples from the conditional density

$$p_{(I,J)}(x) := p(x|\forall i \in I, \forall j \in J, y_i = 1, y_j = 0), \tag{2}$$

for $(I, J) \in \mathcal{I}$ and $\pi_{(I,J)} := p(\forall i \in I, \forall j \in J, y_i = 1, y_j = 0) > 0$. In this work, we propose adding a mild constraint which will allow our sampling algorithm to derive the target density.

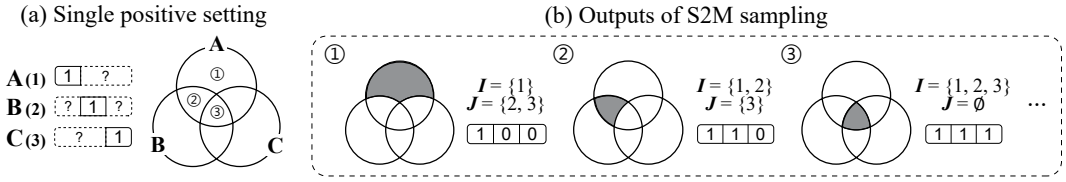


Figure 3: (a) A dataset with single positive labels is given. (b) S2M sampling can draw samples as multi-label data with two index sets: intersection (I) and difference (J).

Assumption 1. *For every $i, j \in N$ such that $i \neq j$, if $p(y_i = 1, y_j = 0) > 0$ and $p(y_j = 1, y_i = 0) > 0$, then* $\operatorname{supp} p(x|y_i = 1, y_j = 0) \cap \operatorname{supp} p(x|y_j = 1, y_i = 0) = \emptyset$.

Assumption 1 states that no data points are likely to be assigned two mutually exclusive labels, which can be naturally assumed in many practical situations. Figure 3 illustrates our problem setting.

3.2 Mining Multi-Label Data with S2M Sampling

Naturally, a question may arise as to whether the supervision given to us is sufficient to obtain multi-label samples. To gain insight into this, we derive a useful theorem which provides an alternative formulation for the target density (2).

Theorem 1. *Let $\{f_{(I,J)} : X \to [0, \infty)\}_{(I,J)\in\mathcal{I}}$ be an indexed family of non-negative measurable functions on X, and let $f_k := f_{(\{k\},\emptyset)}$. Then, the following conditions hold:*

(a) $\forall (I, J) \in \mathcal{I}, f_{(I,J)} = \sum_{S:I\subseteq S, J\subseteq N\setminus S} f_{(S,N\setminus S)}$
(b) $\forall i, j \in N$ *s.t.* $i \neq j, \operatorname{supp} f_{(\{i\},\{j\})} \cap \operatorname{supp} f_{(\{j\},\{i\})} = \emptyset$

if and only if, for every $(I, J) \in \mathcal{I}$,

$$f_{(I,J)} = \begin{cases} (\min_{i\in I} f_i - \max_{j\in J} f_j)^+ & \text{if } J \neq \emptyset \\ \min_{i\in I} f_i & \text{otherwise} \end{cases}, \tag{3}$$

where $(\cdot)^+$ represents the positive part.

Proof. Please refer to Appendix A. □

Let $f_{(I,J)}(x) = p(x, \forall i \in I, \forall j \in J, y_i = 1, y_j = 0)$ for every $(I, J) \in \mathcal{I}$. Then, both (a) and (b) in Theorem 1 hold by the Assumption 1. According to the Theorem 1, if $\pi_{(I,J)} > 0$,

$$p_{(I,J)}(x) = \pi^{-1}_{(I,J)}(\min\{\pi_i p(x|y_i = 1) : i \in I\} - \max\{\pi_j p(x|y_j = 1) : j \in J\} \cup \{0\})^+. \tag{4}$$

The alternative formula (4) shows that the density of the joint class can be derived from the class-conditional densities of the single positive labels. Despite their clear relationship, training a generator that can model the density $p_{(I,J)}$ is a non-trivial problem since the formula consists of several implicitly defined conditional densities, class priors, and variable sets. To address this issue, we propose the application of sampling approaches upon existing GANs. Interestingly, our sampling framework called S2M sampling allows not only for samples to be drawn from the target density, but also the modification of I, J, and class priors at inference time. The rest of this section introduces the main approaches of our S2M sampling method.

Density Ratio Estimation. We utilize several classification networks to compute implicitly defined density ratios. For simplicity, we denote G as a pretrained generator for both unconditional and conditional GANs. G produces data x by taking a latent z and a class label c for class-conditional generation and only z for unconditional generation. We consider three classifiers D_v, D_r, and D_f which are obtained by minimizing $\mathcal{L}_v, \mathcal{L}_r$, and $\mathcal{L}_f$, respectively, *i.e.*,

$$\begin{aligned} \mathcal{L}_v &= -\mathbb{E}_{(x,c)\sim p_{data}(x,c)}[\log D_v(x)] - \mathbb{E}_{x\sim p_G(x)}[\log(1 - D_v(x))], \\ \mathcal{L}_r &= -\mathbb{E}_{(x,c)\sim p_{data}(x,c)}[\log D_r(c|x)],\ \mathcal{L}_f = -\mathbb{E}_{(x,c)\sim p_G(x,c)}[\log D_f(c|x)], \end{aligned} \tag{5}$$

where p_G is the distribution of the generated samples by G. The optimal classifiers trained by these losses D^*_v, D^*_r, and D^*_f satisfy the following equations: $D^*_v(x) = p_{data}(x)/(p_{data}(x) + p_G(x)), D^*_r(c|x) = p(x|y_c = 1)p_{data}(c)/p_{data}(x), D^*_f(c|x) = p_G(x|c)p_G(c)/p_G(x)$. From D^*_v,

D_r^*, and D_f^*, we can access the density ratios $p_{data}(x)/p_G(x)$, $p(x|y_c = 1)/p_{data}(x)$, and $p_G(x|c)/p_G(x)$ which will be used to compute the acceptance probability of the MCMC method.

S2M Sampling for Unconditional GANs. We apply Metropolis-Hastings (MH) independence sampling [27, 21] to draw samples from the complex target distribution p_t. The MH algorithm uses a Markov process where each transition from a current state x_k to the next state x_{k+1} is made by an accept-reject step. At each step of MH independent sampling, a new proposal x' is sampled from a proposal distribution $q(x'|x) = q(x')$ and is then accepted with probability $\alpha(x', x)$ which is given by $\alpha(x', x) = \min\left(1, \frac{p_t(x')q(x)}{p_t(x)q(x')}\right)$. We set $x_{k+1} = x'$ if x' is accepted, and $x_{k+1} = x_k$ otherwise. Under mild conditions, the chain $x_{1:K}$ converges to the unique stationary distribution p_t as $K \to \infty$[1]

Let G be a generator where the support of p_G contains that of $p_{(I,J)}$. To draw samples from $p_{(I,J)}$, we let $p_t = p_{(I,J)}$ and take multiple samples from G as independent proposals, i.e. $q(x'|x) = p_G(x')$. Then, the desired acceptance probability $\alpha(x', x)$ can be calculated using D_v^* and D_r^*:

$$r_{(I,J)}(x) \coloneqq \left(\min\left\{\frac{\pi_i}{p_{data}(i)} D_r^*(i|x) : i \in I\right\} - \max\left\{\frac{\pi_j}{p_{data}(j)} D_r^*(j|x) : j \in J\right\} \cup \{0\}\right)^+,$$
$$\alpha(x', x) = \min\left(1, \frac{p_{(I,J)}(x')/p_G(x')}{p_{(I,J)}(x)/p_G(x)}\right) = \min\left(1, \frac{r_{(I,J)}(x')(D_v^*(x)^{-1} - 1)}{r_{(I,J)}(x)(D_v^*(x')^{-1} - 1)}\right). \tag{6}$$

Different from Turner *et al.* [21], the term $r_{(I,J)}(x')/r_{(I,J)}(x)$ is added to the acceptance probability formula to draw multi-label samples. To obtain uncorrelated samples, a sample is taken after a fixed number of iterations for each chain. The sampling approach allows one to control the parameters I, J, and $\gamma_k \coloneqq \pi_k/p_{data}(k)$ without any additional training of the model. A summary of our S2M sampling algorithm is provided in Appendix B.1.

S2M Sampling for Conditional GANs. cGANs can provide a proposal distribution close to the target distribution $p_{(I,J)}$, which greatly increases the sample efficiency of the MCMC method. Let c be a class label such that the support of class-conditional density $p_G(\cdot|c)$ contains that of $p_{(I,J)}$. At each step of the MH algorithm, the proposal $x' \sim q(x'|x) = p_G(x'|c)$ is accepted with a probability $\alpha_c(x', x)$. The desired $\alpha_c(x', x)$ can be calculated as

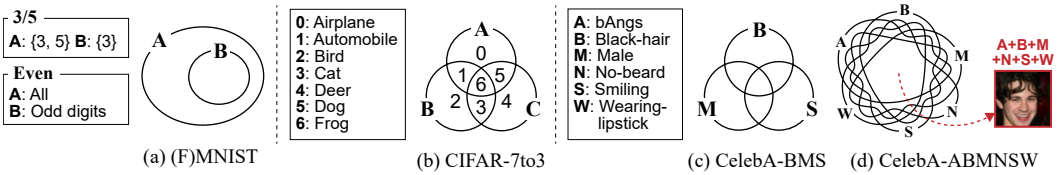
$$\alpha_c(x', x) = \min\left(1, \frac{p_{(I,J)}(x')/p_G(x'|c)}{p_{(I,J)}(x)/p_G(x|c)}\right) = \min\left(1, \frac{r_{(I,J)}(x')D_f^*(c|x)(D_v^*(x)^{-1} - 1)}{r_{(I,J)}(x)D_f^*(c|x')(D_v^*(x')^{-1} - 1)}\right). \tag{7}$$

That is, the sampling method can be adopted to cGANs by additionally computing $D_f^*(c|x)/D_f^*(c|x')$.

Latent Adaptation in S2M Sampling. While our S2M sampling method allows us to draw multi-label samples, the algorithm rejects certain samples through the sampling procedure if the target distribution $p_{(I,J)}$ is significantly different from the generator distribution. Specifically, an independent MH algorithm takes time inversely proportional to the probability of a generated sample belonging to the target class. To alleviate this issue, we propose a technique to improve the sample efficiency from past sampling attempts. Initially, a certain number of pilot target class samples $x_{1:m}$ are drawn using S2M sampling, and then the corresponding target latent samples $t_{1:m}$; $x_k = G(t_k)$ are obtained. Since the latent samples are nearly restricted to the latent prior of GANs (*e.g.*, Gaussian distribution), the distribution $\tilde{p}_z$ of the target latent can be roughly estimated by fitting a simple probabilistic model using $t_{1:m}$. Let $\tilde{p}_G$ be the newly obtained generator distribution induced by G, *i.e.*, $G(z) = x \sim \tilde{p}_G(x)$ for $z \sim \tilde{p}_z(z)$. The sample efficiency can be further improved by using $\tilde{p}_G$ as the proposal distribution in the MH algorithm since $\tilde{p}_G$ is much close to the target distribution. To run the MH algorithm without retraining the classifiers, we approximate $p_G(G(z))/\tilde{p}_G(G(z)) \approx C \cdot p_z(z)/\tilde{p}_z(z)$[2] for some constant C. Then, at each step of the MH algorithm, the proposal $x' \sim q(x'|x) = \tilde{p}_G(x')$ is accepted

[1] For example, the chain of samples is uniformly ergodic if p_t/q is bounded [28].

[2] The equation is derived in the previous latent sampling approaches by the reverse lemma of rejection sampling [25] or the change-of-variables formula [29].



(a) (F)MNIST (b) CIFAR-7to3 (c) CelebA-BMS (d) CelebA-ABMNSW

Figure 4: Experimental settings for (F)MNIST, CIFAR-7to3, CelebA-BMS, and CelebA-ABMNSW.

with a probability $\tilde{\alpha}(x', x)$ which can be calculated as

$$\tilde{\alpha}(x', x) = \min\left(1, \frac{p_{(I,J)}(x')/\tilde{p}_G(x')}{p_{(I,J)}(x)/\tilde{p}_G(x)}\right) \approx \min\left(1, \frac{r_{(I,J)}(x')(D_v^*(x)^{-1} - 1)p_z(z')/\tilde{p}_z(z')}{r_{(I,J)}(x)(D_v^*(x')^{-1} - 1)p_z(z)/\tilde{p}_z(z)}\right), \tag{8}$$

where $x' = G(z')$ and $x = G(z)$. Here, we can explicitly compute the density ratio between p_z and $\tilde{p}_z$ by letting $\tilde{p}_z$ be a known probability density function such as a Gaussian mixture. In practice, if the target class samples rarely appear in the generated samples, one can consider applying latent adaptation repeatedly. For instance, to draw *Black-haired man* samples, we can first search the latent space of *Man* by using latent adaptation, and then perform it in that space again to search the space of *Black-haired man*. In terms of time complexity, this is performed more efficiently than searching the space of *Black-haired man* at once. A detailed description of latent adaptation and a discussion about the time complexity are provided in Appendix B.2 and Appendix C, respectively.

Practical Considerations. We employ three classifiers D_v, D_r, and D_f, to compute the acceptance probability used in the MCMC method. For better training efficiency of the classifiers, we use shared layers, except for the last linear layer. Since the classifiers are not optimal in practice, we scale the temperature of classifiers [30] and adjust γ_k to calibrate the sampling algorithm. Detailed settings and the ablation study are provided in Appendix D and Appendix E.4, respectively.

4 Experiments

In this section, we validate that our S2M sampling method properly draws samples within the joint classes. Specifically, we mainly consider two cases in the single positive setting: (i) one class is entirely contained in another class, (ii) multiple classes are partially overlapping with each other. In the former setting, we verify that S2M sampling can completely filter out samples of a smaller class, and compare our method to the existing models including GenPU [17] and RumiLSGAN [18] on MNIST and Fashion-MNIST (FMNIST). In the latter setting, all possible joint classes are assessed with our S2M sampling method. Subsequently, we evaluate how accurately S2M sampling draws samples in these classes compared to CP-GAN [19] and the models trained with a fully annotated multi-label dataset.

To verify whether a method can accurately generate samples for each joint class, we evaluate *accuracy* which indicates how many generated images are assigned to the target joint classes by a classifier trained with fully annotated data (*i.e.*, multi-label data). Since accuracy itself cannot fully evaluate the distance between the distribution of generated samples and that of real samples, we additionally introduce various metrics to evaluate fidelity (*i.e.*, how realistic the generated samples are) and diversity of generated images: (i) *Inception Score* (IS) [31], (ii) *Fréchet Inception Distance* (FID) [32], (iii) *improved precision and recall* [33], and (iv) *density and coverage* [34].

FID and IS are metrics to evaluate the fidelity and the diversity of generated samples using the features of the pretrained Inception-V3 network [35]. A lower FID and a higher IS indicate higher fidelity and diversity of generated samples. Precision and density measure the ratio of generated samples that falls within the manifold of real samples, so that they evaluate the fidelity of generated samples. Contraily, recall and coverage measure the ratio of real samples that falls within the manifold of the generated samples, and thus evaluating their diversity. That is, a higher value of these metrics indicates that the generated samples have a distribution similar to the real one.

Since several metrics for fidelity and diversity are not applicable for non-ImageNet-like images (*e.g.*, MNIST and FMNIST), we instead use the FID score computed from activations of LeNet5 [36] and denote this as FID†. The test set is used as a reference dataset for evaluating FID. All quantitative results are averaged over three independent trials, and the standard deviation is denoted by subscripts.

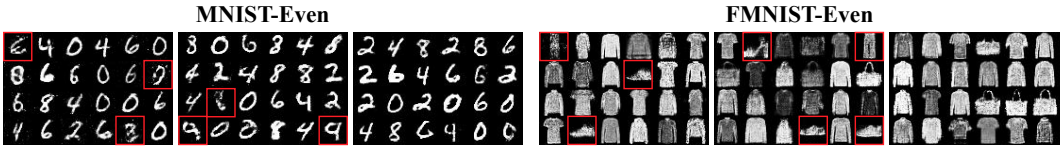


Figure 5: Qualitative results on MNIST-Even and FMNIST-Even.

The qualitative results are randomly sampled in all experiments. Detailed experimental settings, such as architectures and hyperparameters, are provided in Appendix D.

4.1 Sampling for Classes with Inclusion

Experimental Settings. We consider a special case of our problem setting, where one class is entirely contained in another class ($B \subset A$) as shown in Figure 4 (a). This setting is similar to the positive-unlabeled setting [37, 38] where the positive data is included in the unlabeled data. Under this constraint, GenPU [17], RumiLSGAN [18], and CP-GAN [19] can be used to generate samples from the non-overlapping class ($A \setminus B$). Along with these models, we attempt to draw samples from the non-overlapping class by adopting S2M sampling to unconditional WGAN-GP [39]. These experiments were conducted in three settings (See Figure 4): (i) **MNIST-3/5** (A={3, 5}, B={3}), (ii) **MNIST-Even** (A={0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, B={1, 3, 5, 7, 9}), and (iii) **FMNIST-Even** (A={$0_{\text{T-shirt/Top}}$, 1_{Trouser}, 2_{Pullover}, 3_{Dress}, 4_{Coat}, 5_{Sandal}, 6_{Shirt}, 7_{Sneaker}, 8_{Bag}, $9_{\text{Ankle boot}}$}, B={$1, 3, 5, 7, 9$}). For S2M sampling, samples are obtained at 100 MCMC iterations.

Quantitative Results. Table 1 shows the results of our S2M sampling method and the baselines for the non-overlapping classes. The performance of GenPU is reported only for MNIST-3/5 due to its mode collapse issue [40, 41]. S2M sampling adopted to unconditional GAN shows promising results in terms of both accuracy and FID†. In fact, our results are superior to the existing methods specially designed for generating non-overlapping data such as GenPU and RumiLSGAN. Notably, S2M sampling improves accuracy by 8.5% and 6.8% while decreasing FID† compared to the second-best models on MNIST-Even and FMNIST-Even, respectively. This indicates that S2M sampling accurately samples the images of the non-overlapping classes without being biased to a specific mode.

Table 1: Results for different models on MNIST and FMNIST.

Method	MNIST-3/5		MNIST-Even		FMNIST-Even	
	Acc. (%) (↑)	FID† (↓)	Acc. (%) (↑)	FID† (↓)	Acc. (%) (↑)	FID† (↓)
GenPU [17]	99.33±0.56	1.93±1.10	-	-	-	-
RumiLSGAN [18]	77.06±1.54	13.20±1.19	86.11±4.83	3.44±1.39	91.07±0.88	3.23±2.48
CP-GAN [19]	66.89±1.46	19.50±0.90	87.87±0.40	2.23±0.08	81.21±0.67	6.14±0.56
GAN + Ours	**99.52±0.34**	**0.88±0.21**	**96.37±0.25**	**0.86±0.24**	**97.87±0.66**	**2.48±0.29**

Qualitative Results. Figure 5 shows the qualitative results of RumiLSGAN, CPGAN and our method adopted to WGAN-GP. As indicated by the red boxes, RumiLSGAN and CP-GAN fail to completely eliminate the samples of the smaller class (*e.g.*, odd digits for MNIST-Even or *Sneaker* for FMNIST-Even). Conversely, S2M sampling correctly draws samples for the target classes.

4.2 Sampling for Multiple Classes

Experimental Settings. Here, we consider a general case of single positive setting where multiple classes can be overlapping as shown in Figure 4 (b-d). Given n classes, at most $2^n - 1$ joint classes can be obtained. In this setting, we attempt to obtain samples of these joint classes using only single positive labels. We consider cGANs with a projection discriminator (cGAN-PD) [23], AC-GAN [12], and CP-GAN [19] as the baseline generative models.

Since traditional conditional GANs such as cGAN-PD and AC-GAN are not originally designed for generating joint classes, we naively introduce these models in our setting with slight modifications. Concretely, as a method to generate images belonging to n classes, we provide $1/n$ as the conditional value for each class, following Kaneko *et al.* [19]. In our experiments, we expect these models to have a lower bound performance of our results and indicate their results with asterisks (*). In contrast, conditional GANs trained on a fully annotated dataset where all joint classes are specified can be considered as strong baselines in our settings. cGAN-PD and AC-GAN are trained in this setting,

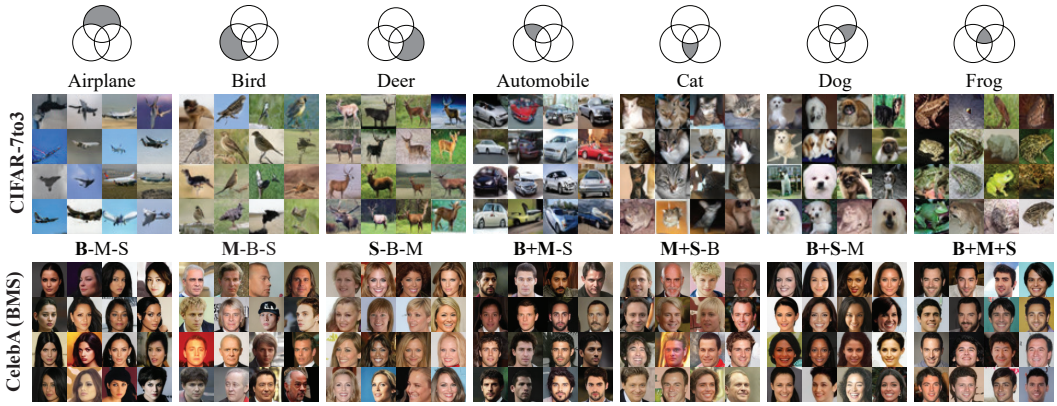


Figure 6: Results of S2M sampling with cGAN-PD on CIFAR-7to3 and CelebA-BMS. The first row depicts the target joint class.

Table 2: Accuracy (%), FID, IS, precision (P), recall (R), density (D), and coverage (C) for different models on real-world datasets.

	Model	Acc. (↑)	FID (↓)	IS (↑)	P (↑)	R (↑)	D (↑)	C (↑)
CIFAR-7to3	cGAN-PD (Oracle) [23]	87.60±0.15	16.55±0.69	8.39±0.18	0.73±0.01	0.65±0.01	0.89±0.05	0.85±0.02
	AC-GAN (Oracle) [12]	94.14±0.39	16.14±0.43	8.44±0.03	0.74±0.00	0.63±0.00	0.89±0.02	0.84±0.00
	cGAN-PD* [23]	27.19±0.53	21.12±0.62	7.94±0.02	0.71±0.01	0.66±0.00	0.81±0.03	0.78±0.01
	AC-GAN* [12]	31.06±1.17	26.42±0.82	7.31±0.10	0.69±0.01	0.66±0.01	0.71±0.01	0.70±0.01
	CP-GAN [19]	42.62±0.54	27.88±2.00	7.42±0.06	0.71±0.00	0.66±0.00	0.76±0.00	0.71±0.02
	GAN + Ours	77.65±1.22	14.42±0.55	8.71±0.16	0.72±0.00	**0.67±0.00**	0.92±0.02	0.87±0.01
	cGAN-PD + Ours	**80.62±2.08**	**14.14±0.34**	**9.06±0.18**	**0.75±0.01**	0.66±0.01	**1.09±0.12**	**0.90±0.01**
CelebA-BMS	cGAN-PD (Oracle) [23]	78.35±1.99	10.38±0.55	2.45±0.07	0.79±0.03	0.48±0.02	1.12±0.18	0.86±0.02
	AC-GAN (Oracle) [12]	91.57±0.43	10.41±1.52	2.40±0.03	0.79±0.02	0.49±0.04	1.18±0.12	0.86±0.03
	cGAN-PD* [23]	39.84±0.70	10.97±1.56	2.32±0.06	0.79±0.01	0.48±0.04	1.19±0.05	0.86±0.02
	AC-GAN* [12]	52.01±1.81	12.39±1.02	2.32±0.10	0.79±0.02	**0.49±0.01**	1.17±0.09	0.85±0.02
	CP-GAN [19]	87.36±2.05	13.38±1.36	2.43±0.09	0.78±0.01	0.45±0.02	1.08±0.04	0.82±0.02
	GAN + Ours	85.22±4.27	**10.50±0.97**	**2.51±0.08**	0.83±0.02	0.48±0.03	1.36±0.08	**0.89±0.02**
	cGAN-PD + Ours	**90.44±1.05**	10.63±0.29	2.46±0.06	**0.84±0.00**	0.48±0.02	**1.40±0.02**	0.88±0.01
CelebA-ABMNSW	cGAN-PD (Oracle) [23]	69.61±2.91	13.54±2.54	2.44±0.11	0.79±0.02	0.42±0.04	1.15±0.10	0.84±0.02
	AC-GAN (Oracle) [12]	91.37±1.14	13.46±1.29	2.30±0.05	0.80±0.02	0.36±0.01	1.17±0.09	0.82±0.01
	cGAN-PD* [23]	15.28±0.08	11.09±0.67	2.32±0.03	0.80±0.01	**0.47±0.03**	1.19±0.07	0.85±0.02
	AC-GAN* [12]	23.97±0.64	11.92±2.06	2.44±0.06	0.77±0.04	0.45±0.02	1.11±0.13	0.82±0.04
	CP-GAN [19]	79.09±1.14	14.97±4.18	2.45±0.04	0.74±0.03	0.46±0.03	0.95±0.08	0.77±0.06
	GAN + Ours	70.04±2.97	**9.77±0.22**	**2.62±0.05**	0.81±0.01	**0.47±0.01**	1.28±0.04	0.87±0.01
	cGAN-PD + Ours	**79.74±0.78**	10.30±0.48	2.45±0.06	**0.85±0.01**	**0.47±0.02**	**1.45±0.06**	**0.88±0.00**

and are denoted as *oracle models*. In the experiments, S2M sampling is adopted on unconditional GAN and cGAN-PD. When adopting S2M sampling to cGAN-PD, a class containing the target joint class is used as the conditional value of cGAN-PD. BigGAN [2] is used as the backbone architecture for all generative models, and only single positive labels are used except for the oracle models.

To demonstrate the effectiveness of our S2M sampling method, we use three real-world image datasets: (i) **CIFAR-7to3** that has three classes (A, B, C), each of which contains four original classes of CIFAR-10, *i.e.* A={*Airplance*, *Automobile*, *Dog*, *Frog*}, B={*Automobile*, *Bird*, *Cat*, *Frog*}, C={*Cat*, *Deer*, *Dog*, *Frog*}, (ii) **CelebA-BMS** that consists of human face images, each of which is labeled for one of three attributes (*Black-hair*, *Male*, and *Smiling*), and (iii) **CelebA-ABMNSW** that consists of human face images, each of which is labeled for one of six attributes (*bAngs*, *Black-hair*, *Male*, *No-beard*, *Smiling*, and *Wearing-lipstick*). As depicted in Figure 4 (b-d), these datasets can have up to 2^3-1 or 2^6-1 joint classes. For S2M sampling, samples are obtained at 200 MCMC iterations for CIFAR-7to3 and CelebA-BMS, and obtained at 1000 MCMC iterations for CelebA-ABMNSW.

Quantitative Results. Table 2 summarizes the quantitative results of joint class generation on CIFAR-7to3, CelebA-BMS, and CelebA-ABMNSW. As expected, we can observe that cGAN-PD* and AC-GAN* cannot accurately generate samples from joint classes, which is consistent with the findings of Kaneko *et al.* [19]. Although CP-GAN shows reasonable accuracy for all experiments, it

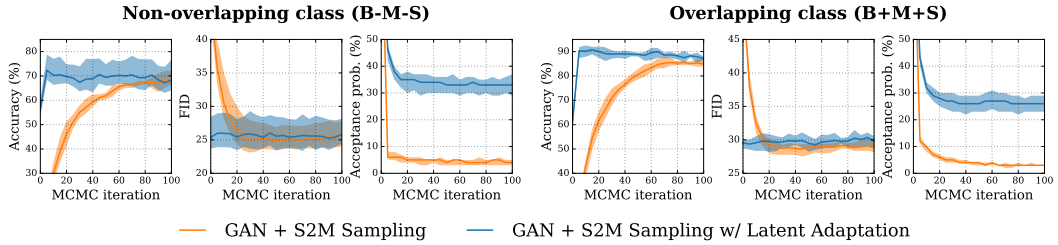


Figure 7: Accuracy (%), FID, and acceptance probability (%) per MCMC iteration for S2M sampling with and without latent adaptation.

suffers from generating diverse samples. As discussed in Section 1, the high FID and low coverage support that CP-GAN tends to generate samples in a narrow scope of real data space.

In all experiments, our S2M sampling method adopted to cGAN-PD consistently outperformes the non-oracle baselines in terms of accuracy and diversity, *e.g.*, in CIFAR-7to3, our method improves accuracy and FID of CP-GAN by 38% and 13.74, respectively. Such improvements verify that our S2M sampling method generates correct samples for joint classes, and also preserves diversity within them. Despite a large sample space of GAN, S2M sampling adopted to unconditional GAN also surpasses non-oracle baselines in terms of both fidelity and diversity while achieving reasonable accuracy. Surprisingly, despite the fact that the oracle models are trained with fully annotated data, S2M sampling shows a comparable performance against those models using only single positive labels. Moreover, as an post-processing method, S2M sampling fully preserves the image quality of GANs and shows the highest fidelity in terms of precision and density.

Qualitative Results. Figure 6 depicts the results of S2M sampling for every possible joint classes on CIFAR-7to3 and CelebA-BMS. The results show that S2M sampling can draw diverse high-quality images from the distributions of all joint classes. Furthermore, S2M sampling is model-agnostic and does not require the retraining of GANs; thus, it can be easily integrated with various GANs including state-of-the-art GANs. We depict more qualitative results of adopting our S2M sampling method to unconditioanl GAN and pretrained StyleGANv2 in Appendix E.5 and Appendix F.

Analysis on Latent Adaptation. To demonstrate the effect of latent adaptation, we tested our latent adaptation technique in a overlapping class (B+M+S) and a non-overlapping class (B-M-S) on the CelebA-BMS dataset. We first perform S2M sampling on unconditional BigGAN in these classes and draw $10k$ pilot latent samples obtained at 100 MCMC iterations. Using these latent samples, a Gaussian mixture of eight components is fit as a new latent of GAN (See Appendix B.2). Due to the discrepancy between the generator distribution and the target distribution, S2M sampling originally shows a low acceptance probability, as shown in Figure 7. In this case, applying latent adaptation increases the acceptance probability and significantly improves both accuracy and FID at the early stage of iterations. This indicates that the generator distribution becomes closer to the target distribution by adopting latent adaptation, resulting in an accelerated convergence speed.

5 Limitations

In general, as the number of attributes increases, the number of each joint class samples in the training dataset are drastically reduced. This may cause two primary problems: (i) training GANs to be able to generate each joint class samples becomes challenging, and S2M sampling cannot draw samples in which GANs do not generate, and (ii) the computation time of the sampling procedure increases. In this case, it may be necessary to apply the latent adaptation technique to shorten MCMC chains. In principle, we can reduce the sampling time complexity to be proportional to the number of attributes (See Appendix C).

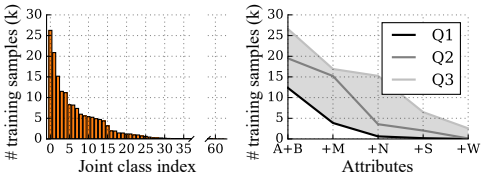


Figure 8: (Left) Distribution of the number of training samples for each joint class on CelebA-ABMNSW. (Right) As attributes are added, the quartiles of the number of samples for each joint class significantly decrease.

As shown in Figure 8, the number of training samples in joint classes rapidly decreases. Because of this, six attributes at most are used on the CelebA dataset for experiments. Almost half of the joint classes have fewer than 100 training samples when using six attributes (*i.e.*, CelebA-ABMNSW). Nevertheless, our method handles a larger number of attributes compared to the existing methods. We believe that our work can be further extended to larger attributes with the sufficient amount of data.

6 Conclusion

In this study, we investigate the single positive setting in the class-conditional generation task and propose a novel sampling framework called S2M sampling. We demonstrate that our proposed S2M sampling method can accurately draw high quality multi-label samples in the single positive setting. To improve sampling efficiency, we also introduce the latent adaptation technique. We believe that our theoretical framework provides a better understanding for weakly-supervised multi-label data generation. Our S2M sampling method has the potential to be applied to a wide variety of generative models as well, including clustering based GANs [42] and diffusion models [43, 44], among others.

Acknowledgments and Disclosure of Funding

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2019-0-00075, Artificial Intelligence Graduate School Program (KAIST) and No. 2021-0-01778, Development of human image synthesis and discrimination technology below the perceptual threshold). This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. NRF-2022R1A2B5B02001913).

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Proc. the International Conference on Learning Representations (ICLR)*, 2019.

[3] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *Proc. the International Conference on Learning Representations (ICLR)*, 2018.

[4] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2019.

[5] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[6] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. of the IEEE international conference on computer vision (ICCV)*, 2017.

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017.

[8] Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018.

[9] Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick. Training region-based object detectors with online hard example mining. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

[10] Christopher Bowles, Liang Chen, Ricardo Guerrero, Paul Bentley, Roger N. Gunn, Alexander Hammers, David Alexander Dickie, Maria del C. Valdés Hernández, Joanna M. Wardlaw, and Daniel Rueckert. GAN augmentation: Augmenting training data using generative adversarial networks. *CoRR*, abs/1810.10863, 2018.

[11] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[12] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proc. the International Conference on Machine Learning (ICML)*, 2017.

[13] Yu-Jing Lin, Po-Wei Wu, Che-Han Chang, Edward Y. Chang, and Shih-Wei Liao. Relgan: Multi-domain image-to-image translation via relative attributes. In *Proc. of the IEEE international conference on computer vision (ICCV)*, 2019.

[14] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2014.

[15] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2019.

[16] Elijah Cole, Oisin Mac Aodha, Titouan Lorieul, Pietro Perona, Dan Morris, and Nebojsa Jojic. Multi-label learning from single positive labels. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2021.

[17] Ming Hou, Brahim Chaib-draa, Chao Li, and Qibin Zhao. Generative adversarial positive-unlabelled learning. In *Proc. the International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

[18] Siddarth Asokan and Chandra Sekhar Seelamantula. Teaching a GAN what not to learn. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[19] Takuhiro Kaneko, Yoshitaka Ushiku, and Tatsuya Harada. Class-distinct and class-mutual image generation with gans. In *Proc. of the British Machine Vision Conference (BMVC)*, 2019.

[20] Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian J. Goodfellow, and Augustus Odena. Discriminator rejection sampling. In *Proc. the International Conference on Learning Representations (ICLR)*, 2019.

[21] Ryan D. Turner, Jane Hung, Eric Frank, Yunus Saatchi, and Jason Yosinski. Metropolis-hastings generative adversarial networks. In *Proc. the International Conference on Machine Learning (ICML)*, 2019.

[22] Mingming Gong, Yanwu Xu, Chunyuan Li, Kun Zhang, and Kayhan Batmanghelich. Twin auxiliary classifiers GAN. *CoRR*, abs/1907.02690, 2019.

[23] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. In *Proc. the International Conference on Learning Representations (ICLR)*, 2018.

[24] Minguk Kang and Jaesik Park. Contragan: Contrastive learning for conditional image generation. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[25] Tong Che, Ruixiang Zhang, Jascha Sohl-Dickstein, Hugo Larochelle, Liam Paull, Yuan Cao, and Yoshua Bengio. Your GAN is secretly an energy-based model and you should use discriminator driven latent sampling. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[26] Sangwoo Mo, Chiheon Kim, Sungwoong Kim, Minsu Cho, and Jinwoo Shin. Mining GOLD samples for conditional gans. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[27] Luke Tierney. Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4):1701–1728, 1994.

[28] K. L. Mengersen and R. L. Tweedie. Rates of convergence of the Hastings and Metropolis algorithms. *The Annals of Statistics*, 24(1):101 – 121, 1996.

[29] Yifei Wang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Reparameterized sampling for generative adversarial networks. In *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference*, 2021.

[30] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proc. the International Conference on Machine Learning (ICML)*, 2017.

[31] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[32] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[33] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[34] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *Proc. the International Conference on Machine Learning (ICML)*, 2020.

[35] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

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

[37] François Denis. PAC learning from positive statistical queries. In *ALT*, Lecture Notes in Computer Science, 1998.

[38] François Denis, Rémi Gilleron, and Fabien Letouzey. Learning from positive and unlabeled examples. *Theor. Comput. Sci.*, 2005.

[39] Xiang Wei, Boqing Gong, Zixia Liu, Wei Lu, and Liqiang Wang. Improving the improved training of wasserstein gans: A consistency term and its dual effect. In *Proc. the International Conference on Learning Representations (ICLR)*, 2018.

[40] Hui Chen, Fangqing Liu, Yin Wang, Liyue Zhao, and Hao Wu. A variational approach for learning from positive and unlabeled data. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[41] Florent Chiaroni, Ghazaleh Khodabandelou, Mohamed-Cherif Rahal, Nicolas Hueber, and Frédéric Dufaux. Counter-examples generation from a positive unlabeled image dataset. *Pattern Recognition*, 107:107527, 2020.

[42] Arantxa Casanova, Marlène Careil, Jakob Verbeek, Michal Drozdzal, and Adriana Romero-Soriano. Instance-conditioned GAN. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[43] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[44] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[45] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. the International Conference on Learning Representations (ICLR)*, 2015.

[46] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018.

[47] Jae Hyun Lim and Jong Chul Ye. Geometric GAN. *CoRR*, abs/1705.02894, 2017.

[48] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *Proc. the International Conference on Learning Representations (ICLR)*, 2018.

[49] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proc. of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2020.

[50] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 2008.

[51] Pavlin G. Policar, Martin Strazar, and Blaz Zupan. Embedding to reference t-sne space addresses batch effects in single-cell classification. In *Discovery Science - 22nd International Conference, DS*, 2019.

[52] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *CoRR*, abs/1908.09635, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We provide the codes in the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We reported the standard deviation in the experimental results.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No] We used publicly available benchmark datasets: MNIST, FMNIST, CIFAR10, CelebA, and CelebA-HQ.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We provide the codes in the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] The benchmark datasets used in our paper are widely used for research purpose.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] The benchmark datasets used in our paper are widely used by others and do not contain any offensive contents.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Proof of Main Theorem

Let $\{f_{(I,J)} : X \to [0,\infty)\}_{(I,J)\in\mathcal{I}}$ be an indexed family of non-negative measurable functions on X, and let $f_k \coloneqq f_{(\{k\},\emptyset)}$ for $k \in N$. We will consider two following properties:

(a) $\forall (I,J) \in \mathcal{I}, f_{(I,J)} = \sum_{S:I\subseteq S, J\subseteq N\setminus S} f_{(S,N\setminus S)}$

(b) $\forall i,j \in N$ s.t. $i \neq j, \operatorname{supp} f_{(\{i\},\{j\})} \cap \operatorname{supp} f_{(\{j\},\{i\})} = \emptyset$

We first derive some useful lemmas to prove the necessity of Theorem 1.

Lemma 1. *Assume that (a) and (b) hold.* $\forall (I_1,J_1),(I_2,J_2) \in \mathcal{I}$ *s.t.* $I_1 \cap J_2 \neq \emptyset, J_1 \cap I_2 \neq \emptyset$, $\operatorname{supp} f_{(I_1,J_1)} \cap \operatorname{supp} f_{(I_2,J_2)} = \emptyset$.

Proof. Choose any $u \in I_1 \cap J_2, v \in J_1 \cap I_2$. By (a),

$$\begin{aligned} f_{(\{u\},\{v\})} &= \sum\nolimits_{S:u\in S, v\in N\setminus S} f_{(S,N\setminus S)} \geq \sum\nolimits_{S:I_1\subseteq S, J_1\subseteq N\setminus S} f_{(S,N\setminus S)} = f_{(I_1,J_1)} \\ f_{(\{v\},\{u\})} &= \sum\nolimits_{S:v\in S, u\in N\setminus S} f_{(S,N\setminus S)} \geq \sum\nolimits_{S:I_2\subseteq S, J_2\subseteq N\setminus S} f_{(S,N\setminus S)} = f_{(I_2,J_2)}. \end{aligned} \tag{9}$$

If $u = v$, then $I_1 \cap J_1 \neq \emptyset$ which contradicts the definition of $\mathcal{I}$. Therefore, $u \neq v$. By (b), $\operatorname{supp} f_{(I_1,J_1)} \cap \operatorname{supp} f_{(I_2,J_2)} \subseteq \operatorname{supp} f_{(\{u\},\{v\})} \cap \operatorname{supp} f_{(\{v\},\{u\})} = \emptyset$. □

Lemma 2. *Assume that (a) and (b) hold.* $\forall i \in N, \forall I \subseteq N$ *s.t.* $\{i\} \subsetneq I$, $f_{(I\setminus\{i\},\{i\})} \cap \sum_{S:i\in S, I\nsubseteq S\subseteq N} f_{(S,N\setminus S)} = \emptyset$.

Proof. Let $S \subseteq N$ s.t. $i \in S \not\supseteq I$. Then, $\emptyset \neq I \setminus S \subseteq (I\setminus\{i\}) \cap (N\setminus S)$. By Lemma 1, $f_{(I\setminus\{i\},\{i\})} \cap f_{(S,N\setminus S)} = \emptyset$. Therefore, $f_{(I\setminus\{i\},\{i\})} \cap \sum_{S:i\in S, I\nsubseteq S\subseteq N} f_{(S,N\setminus S)} = \emptyset$. □

Lemma 3. *Assume that (a) and (b) hold.* $\forall I \subseteq N$ *s.t.* $I \neq \emptyset$, $f_{(I,\emptyset)} = \min_{i\in I} f_i$.

Proof. We will use induction to prove the lemma. Let $P(k)$ be the following statement.

$$P(k) : \forall I \text{ s.t. } 1 \leq |I| = k \leq |N|, \text{then } f_{(I,\emptyset)} = \min_{i\in I} f_i. \tag{10}$$

For the base case $k = 1$, the statement holds by the definition. Assume that the induction hypothesis for $k \leq l < |N|$ holds. Consider $|I| = l+1$ and choose any $i \in I$. Then,

$$\begin{aligned} \min_{i\in I} f_i &= \min\{f_{(I\setminus\{i\},\emptyset)}, f_{(\{i\},\emptyset)}\} && \text{by the induction hypothesis} \\ &= f_{(I\setminus\{i\},\emptyset)} - \max\{f_{(I\setminus\{i\},\emptyset)} - f_{(\{i\},\emptyset)}, 0\} \\ &= f_{(I\setminus\{i\},\emptyset)} - \max\{f_{(I\setminus\{i\},\{i\})} - \sum\nolimits_{S:i\in S, I\nsubseteq S\subseteq N} f_{(S,N\setminus S)}, 0\} \\ &= f_{(I\setminus\{i\},\emptyset)} - f_{(I\setminus\{i\},\{i\})} && \text{by Lemma 2} \\ &= f_{(I,\emptyset)} \end{aligned} \tag{11}$$

Therefore, $P(l+1)$ holds. We conclude that $f_{(I,\emptyset)} = \min_{i\in I} f_i$ for $\emptyset \neq I \subseteq N$. □

Theorem 1. *Let* $\{f_{(I,J)} : X \to [0,\infty)\}_{(I,J)\in\mathcal{I}}$ *be an indexed family of non-negative measurable functions on* X*, and let* $f_k \coloneqq f_{(\{k\},\emptyset)}$*. Then, the following conditions hold:*

(a) $\forall (I,J) \in \mathcal{I}, f_{(I,J)} = \sum_{S:I\subseteq S, J\subseteq N\setminus S} f_{(S,N\setminus S)}$

(b) $\forall i,j \in N$ *s.t.* $i \neq j, \operatorname{supp} f_{(\{i\},\{j\})} \cap \operatorname{supp} f_{(\{j\},\{i\})} = \emptyset$

if and only if, for every $(I,J) \in \mathcal{I}$,

$$f_{(I,J)} = \begin{cases} (\min_{i\in I} f_i - \max_{j\in J} f_j)^+ & \text{if } J \neq \emptyset \\ \min_{i\in I} f_i & \text{otherwise} \end{cases}, \tag{3}$$

where $(\cdot)^+$ *represents the positive part.*

Proof. We first show the necessity of the condition. Assume that (a) and (b) hold. If $J = \emptyset$, then $f_{(I,J)} = \min_{i \in I} f_i$ by Lemma2. Hence, we may assume that $J \neq \emptyset$. Fix $x \in X$, and let $\{a_1, a_2, ..., a_{|J|}\}$ be an arrangement of J so that $f_{a_i}(x) \leq f_{a_j}(x)$ for all $i < j$. For every $\emptyset \neq S \subseteq J$, we let $m(S)$ denote the minimum index s such that $a_s \in S$.

Note that

$$\begin{aligned} f_{(I,J)}(x) &= \sum_{S \subseteq J} (-1)^{|S|} f_{(I \cup S, \emptyset)}(x) \quad \text{by Inclusion–exclusion principle} \\ &= \sum_{S \subseteq J} (-1)^{|S|} \min_{i \in I \cup S} f_i(x) \quad \text{by Lemma 3} \end{aligned} \tag{12}$$

We now decompose the last summation into three cases.

(i) $S = \emptyset$

$$(-1)^{|S|} \min_{i \in I \cup S} f_i(x) = \min_{i \in I} f_i(x). \tag{13}$$

(ii) $m(S) < |J|$

$$\begin{aligned} \sum_{S: m(S) < |J|} (-1)^{|S|} \min_{i \in I \cup S} f_i(x) &= \sum_{j < |J|} \sum_{S: m(S) = j} (-1)^{|S|} \min_{i \in I \cup \{a_j\}} f_i(x) \\ &= \sum_{j < |J|} \left(\min_{i \in I \cup \{a_j\}} f_i(x) \right) \left\{ (-1) \cdot 2^{|J|-j-1} + 2^{|J|-j-1} \right\} \\ &= 0. \end{aligned} \tag{14}$$

(iii) $m(S) = |J|$

$$(-1)^{|S|} \min_{i \in I \cup S} f_i(x) = - \min_{i \in I \cup \{a_{|J|}\}} f_i(x). \tag{15}$$

Summing up all of the above terms gives the rest result.

$$\begin{aligned} f_{(I,J)}(x) &= \min_{i \in I} f_i(x) - \min_{i \in I \cup \{a_{|J|}\}} f_i(x) \\ &= \min_{i \in I} f_i(x) - \min\{\min_{i \in I} f_i(x), \max_{j \in J} f_j(x)\} \\ &= \left(\min_{i \in I} f_i(x) - \max_{j \in J} f_j(x) \right)^+. \end{aligned} \tag{16}$$

To show the sufficiency, assume

$$\forall (I,J) \in \mathcal{I}, f_{(I,J)} = \begin{cases} (\min_{i \in I} f_i - \max_{j \in J} f_j)^+ & \text{if } J \neq \emptyset \\ \min_{i \in I} f_i & \text{otherwise} \end{cases}. \tag{17}$$

Let us assume that $f_{(I,J)} \neq \sum_{S: I \subseteq S, J \subseteq N \setminus S} f_{(S, N \setminus S)}$ for some $(I,J) \in \mathcal{I}$. Choose such I, J so that $|I| + |J|$ is maximum. Note that $I \cup J \subsetneq N$ because $\sum_{S: I \subseteq S, J \subseteq N \setminus S} f_{(S, N \setminus S)}$ is exactly the same expression as $f_{(I,J)}$ for $I \cup J = N$. Hence, we can choose some $k \in N \setminus (I \cup J)$. By the maximality of $|I| + |J|$, the following two equations hold.

$$\begin{aligned} f_{(I, J \cup \{k\})} &= \sum\nolimits_{S: I \subseteq S, J \cup \{k\} \subseteq N \setminus S} f_{(S, N \setminus S)} \\ f_{(I \cup \{k\}, J)} &= \sum\nolimits_{S: I \cup \{k\} \subseteq S, J \subseteq N \setminus S} f_{(S, N \setminus S)}. \end{aligned} \tag{18}$$

We use above two equations and consider all possible inequalities among $\min_{i\in I} f_i$, $max_{j\in J} f_j$, and f_k. The following equation always holds regardless of these inequalities.

$$\begin{aligned}
\sum_{S:I\subseteq S, J\subseteq N\setminus S} & f_{(S,N\setminus S)} = f_{(I,J\cup\{k\})} + f_{(I\cup\{k\},J)} \\
&= \begin{cases} \left(\min_{i\in I} f_i - \max_{j\in J\cup\{k\}} f_j\right)^+ + \left(\min_{i\in I\cup\{k\}} f_i - \max_{j\in J} f_j\right)^+ & \text{if } J \neq \emptyset \\ (\min_{i\in I} f_i - f_k)^+ + \min_{i\in I\cup\{k\}} f_i & \text{otherwise} \end{cases} \\
&= \begin{cases} (\min_{i\in I} f_i - \max_{j\in J} f_j)^+ & \text{if } J \neq \emptyset \\ \min_{i\in I} f_i & \text{otherwise} \end{cases} \\
&= f_{(I,J)},
\end{aligned} \tag{19}$$

which leads to a contradiction.

Also, for every $i, j \in N$ such that $i \neq j$,

$$\begin{aligned}
\min(f_{(\{i\},\{j\})}, f_{(\{j\},\{i\})}) &= \min\{(f_i - f_j)^+, (f_j - f_i)^+\} \\
&= (\min\{f_i - f_j, f_j - f_i\})^+ \\
&= 0.
\end{aligned} \tag{20}$$

Therefore, (a) and (b) hold. □

B Description for S2M Sampling

B.1 Pseudocode of S2M Sampling

In Section 3.2, we describe how to build S2M sampling upon unconditional GANs and class-conditional GANs. Algorithm 1 illustrates the use of S2M sampling for GANs. This algorithm can be easily modified to the conditional versions by replacing the acceptance probability α (See Equation 7).

Algorithm 1 *S2M Sampling for GANs*

Input: generator G, classifiers D_v^*, D_r^*, intersection index set I, difference index set J, and class prior ratios $\gamma_{1:N}$
Output: filtered sample x

Choose any $x \in \operatorname{supp} p_{(I,J)}$.
for $k = 1$ to K **do**
    Draw x' from G.
    Draw u from Uniform(0,1).
    $r_i \leftarrow \gamma_i D_r^*(i|x)$ for every $i \in I \cup J$
    $r'_i \leftarrow \gamma_i D_r^*(i|x')$ for every $i \in I \cup J$
    $\alpha \leftarrow \min\left(1, \frac{\left(\min\{r'_i : i\in I\} - \max\{r'_j : j\in J\}\cup\{0\}\right)^+ (D_v^*(x)^{-1} - 1)}{(\min\{r_i : i\in I\} - \max\{r_j : j\in J\}\cup\{0\})^+ (D_v^*(x')^{-1} - 1)}\right)$
    if $u \leq \alpha$ **then**
        $x \leftarrow x'$
    end if
end for

B.2 Latent Adaptation with Gaussian Mixture Model

In Section 3.2, we describe the latent adaptation technique to improve the sampling efficiency of our proposed S2M sampling method. In this section, we provide an example for the latent adaptation technique using a Gaussian mixture model. The real examples of using latent adaption are shown in Figure 9. After obtaining target latent samples $t_{1:m}$ from S2M sampling, we use those samples to fit a multivariate Gaussian mixture model $\tilde{p}_z(z) = \sum_{i=1}^{M} \phi_i \mathcal{N}(z|\mu_i, \Sigma_i)$. The parameters can be updated using an expectation–maximization algorithm [3] As explained in Section 3.2, we run the MH

[3] Christopher M. Bishop. Pattern recognition and machine learning, 5th Edition. 2007.

algorithm where the proposal $x' \sim q(x'|x) = \tilde{p}_G(x')$ is accepted with a probability $\tilde{\alpha}(x', x)$ which is calculated as

$$\tilde{\alpha}(x', x) = \min\left(1, \frac{p_{(I,J)}(x')/\tilde{p}_G(x')}{p_{(I,J)}(x)/\tilde{p}_G(x)}\right) \approx \min\left(1, \frac{r_{(I,J)}(x')(D_v^*(x)^{-1} - 1)p_z(z')/\tilde{p}_z(z')}{r_{(I,J)}(x)(D_v^*(x')^{-1} - 1)p_z(z)/\tilde{p}_z(z)}\right). \tag{21}$$

If the latent prior of the generator G is the standard multivariate normal distribution $p_z(z) = \mathcal{N}(z|0, I_d)$, we can compute the acceptance probability by explicitly calculating the density ratio $p_z(z)/\tilde{p}_z(z)$ as follows:

$$\frac{p_z(z)}{\tilde{p}_z(z)} = \frac{\mathcal{N}(z|\mu_0, \Sigma_0)}{\sum_{i=1}^M \phi_i \mathcal{N}(z|\mu_i, \Sigma_i)} = \left(\sum_{i=1}^M \phi_i |\Sigma_i|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(z-\mu_i)^T {\Sigma_i}^{-1}(z-\mu_i) + \frac{1}{2}z^T z\right)\right)^{-1}. \tag{22}$$

In our experiments, some determinants of the covariance matrices often approach zero, resulting in numerical errors. One way to readily avoid this problem is to let $\tilde{p}_z$ be a Gaussian mixture with a shared covariance matrix Σ, *i.e.*, $\Sigma_1 = \Sigma_2 = ... = \Sigma_M = \Sigma$. Then, we no longer need to compute the determinants of the covariance matrices to calculate the acceptance probability because $p_z(z')\tilde{p}_z(z)/\tilde{p}_z(z')/p_z(z)$ is simplified as follows:

$$\frac{p_z(z')\tilde{p}_z(z)}{\tilde{p}_z(z')p_z(z)} = \frac{\sum_{i=1}^M \phi_i \exp\left(-\frac{1}{2}(z-\mu_i)^T\Sigma^{-1}(z-\mu_i) + \frac{1}{2}z^T z\right)}{\sum_{i=1}^M \phi_i \exp\left(-\frac{1}{2}(z'-\mu_i)^T\Sigma^{-1}(z'-\mu_i) + \frac{1}{2}z'^T z'\right)}. \tag{23}$$

Algorithm 2 illustrates the sampling process using the Gaussian mixture latent with a shared covariance matrix. After collecting the filtered latent samples $z_{1:s}$ through Algorithm 2, we can get target class samples $x_{1:s}$ by taking $x_i = G(z_i)$ for $i = 1, 2, ..., s$. As explained in Section 3.2, the latent adaptation technique can be applied iteratively by increasing the number of specified attributes one by one. Algorithm 3 illustrates this strategy and Section C provides the computation complexity analysis of the algorithm.

Algorithm 2 *Sampling with Adapted Latent Space*

Input: generator G, classifiers D_v^*, D_r^*, intersection index set I, difference index set J, class prior ratios $\gamma_{1:N}$ and target latent distribution parameters $\phi_{1:M}, \mu_{1:M}, \Sigma$
Output: filtered latent sample z

Let $\tilde{p}_z(z) := \sum_{i=1}^M \phi_i \mathcal{N}(z|\mu_i, \Sigma)$.
Choose any z such that $G(z) \in \operatorname{supp} p_{(I,J)}$.
for $k = 1$ to K **do**
    Draw z' from $\tilde{p}_z$.
    Draw u from Uniform(0,1).
    $r_i \leftarrow \gamma_i D_r^*(i|G(z))$ for every $i \in I \cup J$
    $r'_i \leftarrow \gamma_i D_r^*(i|G(z'))$ for every $i \in I \cup J$
    $d \leftarrow \frac{\sum_{i=1}^M \phi_i \exp\left(-\frac{1}{2}(z-\mu_i)^T\Sigma^{-1}(z-\mu_i)+\frac{1}{2}z^T z\right)}{\sum_{i=1}^M \phi_i \exp\left(-\frac{1}{2}(z'-\mu_i)^T\Sigma^{-1}(z'-\mu_i)+\frac{1}{2}z'^T z'\right)}$
    $\alpha \leftarrow \min\left(1, \frac{\left(\min\{r'_i : i \in I\} - \max\{r'_j : j \in J\} \cup \{0\}\right)^+ (D_v^*(G(z))^{-1} - 1)}{(\min\{r_i : i \in I\} - \max\{r_j : j \in J\} \cup \{0\})^+ (D_v^*(G(z'))^{-1} - 1)} \cdot d\right)$
    if $u \leq \alpha$ **then**
        $z \leftarrow z'$
    end if
end for

Algorithm 3 *Repeating Latent Adaptation*

Input: generator G, classifiers D_v^*, D_r^*, intersection index set I, difference index set J, and class prior ratios $\gamma_{1:N}$
Output: filtered latent samples $z_{1:s}$

Let $I_0 \subseteq I$ such that $|I_0| = 1$, and let $J_0 = \emptyset$.
$t_{1:m} \leftarrow$ MHAlgorithm$(G, D_v^*, D_r^*, I_0, J_0, \gamma_{1:N})$ ▷ Obtain latent samples using Algorihtm 1
while $I_0 \cup J_0 \neq I \cup J$ **do**
    Fit $\tilde{p}_z(z) = \sum_{i=1}^{M} \phi_i \mathcal{N}(z|\mu_i, \Sigma)$ with $t_{1:m}$ using the expectation-maximization algorithm.
    Choose $\alpha \subseteq I \setminus I_0, \beta \subseteq J \setminus J_0$ such that $|\alpha \cup \beta| = 1$.
    $I_0 \leftarrow I_0 \cup \{\alpha\}$
    $J_0 \leftarrow J_0 \cup \{\beta\}$
    $t_{1:m} \leftarrow$ AdaptiveLatentSampling$(G, D_v^*, D_r^*, I_0, J_0, \gamma_{1:N}, \phi_{1:M}, \mu_{1:M}, \Sigma)$ ▷ Obtain latent samples using Algorihtm 2
end while
Fit $\tilde{p}_z(z) = \sum_{i=1}^{M} \phi_i \mathcal{N}(z|\mu_i, \Sigma)$ with $t_{1:m}$ using the expectation-maximization algorithm.
$z_{1:s} \leftarrow$ AdaptiveLatentSampling$(G, D_v^*, D_r^*, I_0, J_0, \gamma_{1:N}, \phi_{1:M}, \mu_{1:M}, \Sigma)$ ▷ Obtain latent samples using Algorihtm 2

(1) GAN (Original) (2) GAN w/ **latent adaptation** (3) **S2M sampling** (10 steps)

Figure 9: Results of sequentially applying latent adaptation and S2M sampling for *B+M+S* joint class. (1) Before applying latent adaptation, GANs draw diverse CelebA images. (2) After applying latent adaptation to B+M+S class, the generator produces images of that class with a high probability. (3) By using latent adaptation, S2M sampling can remove samples outside of that class even with a few MCMC iterations.

C Computational Complexity Analysis

Inference time of different architectures. In this section, we compare the inference time of several architectures as shown in Table 3. The classifiers and Gaussian mixture sampler add a small amount of time to the inference process for S2M sampling. To account for these components, we measure the inference time per iteration during the sampling procedure. We use a single RTX 3090 GPU for this experiment.

Table 3: Inference time (ms) of several architectures on CelebA-BMS. PD denote the projection discriminator inference, CLS denote the classifier inference, and GMM denote the gaussian mixture sampler inference. We use the batch size of 64.

Method	CP-GAN	BigGAN	BigGAN + PD	BigGAN + CLS	BigGAN + CLS + GMM
Time (ms)	31.61	29.56	31.66	35.86	36.18

Time complexity of sampling algorithms. To simplify the analysis, we assume that a generator distribution is close to a real distribution, and let α be the probability of a generated sample belonging to a target joint class. Then, we have $\frac{p_{(I,J)}(x)}{p_G(x)} \leq \frac{1}{\alpha}$ for all x. By Theorem 2.1 of Mengersen *et*

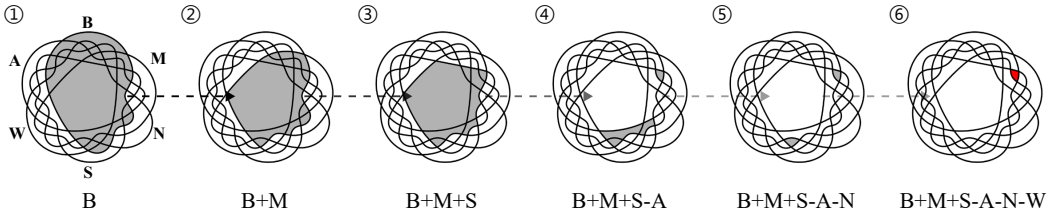


Figure 10: Our latent adaptation can be repeatedly applied for the efficiency of S2M sampling. For instance, to generate images of *B+M+S-A-N-W* joint class (6) with S2M sampling, we can first search the space of *Black-hair* attribute (1) and then gradually increase the number of attributes.

al. [28] the convergence speed of the independent MH algorithm in this condition is given by

$$||P^t(x,\cdot) - p_{(I,J)}||_{TV} \leq (1-\alpha)^t, \tag{24}$$

where $||\cdot||_{TV}$ is the total variation distance and $P^t(x,\cdot)$ is the t-step transition probability density for an initial state x. To analyze the inference time complexity, we now compute the number of MCMC iterations required until the distance is small enough. For a given fixed small ϵ, the number of steps required for the distance to fall within ϵ is $t = \frac{\ln \epsilon}{\ln(1-\alpha)} = O(\frac{1}{\alpha})$. Intuitively, the sampling algorithm without latent adaptation (Algorithm 1) takes time inversely proportional to the relative ratio of the target class samples out of the entire dataset used to train the generator.

Given a fixed number of data, as the number of attributes increases, α decreases, and the algorithm converges slowly. To alleviate such inefficiency, we additionally proposed latent adaptation which collects a certain amount of target class samples using Algorithm 1 and then uses it fit to a new generator distribution close the target distribution. If the newly obtained generator distribution is sufficiently closed to the target distribution, the sampling algorithm using the new generator will take a constant time to produce the target class samples. The remaining question is how long it take to apply latent adaptation.

To discuss about the time complexity of latent adaptation scale with the number of attributes, we let q_i be the probability of a generated sample belonging to the i-th class and assume conditional independence among the classes, *i.e.*, $p(y_i|x, y_1, ..., y_{i-1}, y_{i+1}, ..., y_n) = p(y_i|x)$. Let m be the number of latent samples used to fit the new proposal distribution and c be the overhead introduced by fitting the latent distribution. We consider two scenarios of applying latent adaptation and analyze the time complexity taken to completely fit the latent space in each scenario: (i) Applying latent adaptation by searching the latent space of target joint class samples at once. (ii) Applying latent adaptation repeatedly by increasing the number of specified attributes one by one as shown in Figure 10 (See Algorithm 3 for details). For the case (i), the algorithm runs the MH algorithm once to draw the target class samples and the probability of each generated sample belonging to that class is $\prod_{i\in I} q_i \prod_{j\in J}(1-q_j)$. Hence, it takes $O(m(\prod_{i\in I} q_i \prod_{j\in J}(1-q_j))^{-1} + c)$. For the case (ii), the algorithm runs the MH algorithm once for each class of I and each class of J, so it takes $O(m(\sum_{i\in I} q_i^{-1} + \sum_{j\in J}(1-q_j)^{-1}) + c(|I|+|J|))$. That is, Algorithm 3 can run efficiently as it takes time linearly proportional to the number of attributes. When the number of attributes is small, (i) is also a good way to use latent adaptation, since it has a small overhead of fitting the latent distribution.

To validate the empirical effectiveness of applications of latent adaptation, we select three joint classes of CelebA-ABMNSW, whose ratio of training samples is lower than 3%. For the method (i), sampling for obtaining latent samples is performed by 90 MCMC iterations. For the method (ii), the latent sampling is performed by 15 MCMC iterations for each attribute. For both cases, we collect $10K$ latent samples to perform latent adaptation. After fitting a new latent distribution suit for a target joint class, we again perform a sampling algorithm to draw target joint class samples. We measure the number of MCMC iterations required until accuracy converges. As shown in Table 4, latent adaptation drastically shortens the MCMC chain for all cases, and the method (ii) requires fewer MCMC iterations to draw the target joint class samples than the method (i).

Table 4: Comparison of applications of latent adaptation on CelebA-ABMNSW. For each joint class, we denote the ratio of training samples of the class in the parentheses. "-" in LA type indicates that latent adaptation is not applied. # of searching iterations refers to the total number of MCMC steps performed to fit the latent distribution. # of sampling iterations refers to the number of MCMC steps required until accuracy converges. Accuracy and FID are measured at this step.

Class (Ratio)	LA type	# of searching iterations	# of sampling iterations	Accuracy	FID
B+M+S-A-N-W (1.95%)	-	0	320	53.51%	44.72
	(i)	90	20	57.87%	45.37
	(ii)	90	15	59.08%	46.41
B+M+N+S-A-W (2.68%)	-	0	165	43.78%	38.64
	(i)	90	35	42.70%	38.34
	(ii)	90	5	44.14%	37.15
A+N+W-B-M-S (2.89%)	-	0	335	84.72%	32.03
	(i)	90	40	86.20%	32.22
	(ii)	90	5	86.94%	33.89

D Experiments Details

D.1 MNIST and FMNIST

Each of MNIST and FMNIST dataset consists of a training set of $60k$ images and a test set of $10k$ images. We use 10% of the training set as the validation set. To make the training dataset annotated by single positive labels, we distribute the images belong to the joint classes equally to each corresponding classes of the single positive labels. We evaluate accuracy on MNIST and FMNIST dataset using LeNet5 [36] trained with fully annotated dataset.

The generative models for MNIST and FMNIST are discussed in Section 4.1. As similar to the original setting of GenPU, the generator consists of ReLU activations and fully connected layers of input size: 100-256-256-784. The discriminator consists of ReLU activations and fully connected layers of input size: 784-256-256. As for the GAN objective, we follow the settings introduced by the authors for baselines, and use WGAN-GP [39] for our model. We train all generative models using Adam optimizer [45] with a learning rate of 0.0001, $\beta_1 = 0.5, \beta_2 = 0.999$, and a batch size of 64. The generator is trained for $200k$ iterations, and two updates of the discriminator are performed for every update of the generator.

Classification networks used for S2M sampling are obtained from multiple branches of LeNet5 architecture. We train the classifier using Adam optimizer. For MNIST 3/5 dataset, the classifier is trained for 10 epochs with a learning rate of 0.001, and the temperature of D_r is set to 2. For MNIST and FMNIST Even dataset, the classifier is trained for 50 epochs with a learning rate of of 0.0001, and the temperature of D_v is set to 4. Each γ_k corresponding to the intersection set is set to 0.1 for both MNIST and FMNIST.

D.2 CIFAR-10 and CelebA

For CIFAR-10 dataset, we use 10% of the training set as the validation set. We follow the original partition description and resize images to 64×64 for training efficiency on CelebA dataset. To make the training datasets annotated by single positive labels, we distribute the images belong to the joint classes equally to each corresponding classes of the single positive labels. We evaluate accuracy on CIFAR-10 and CelebA dataset using MobileNet V2 [46] trained with fully annotated dataset.

The generative models for CIFAR-7to3 and CelebA datasets are discussed in Section 4.2. We use BigGAN [2] architecture for all generative models and follow the PyTorch implementation[4]. We use hinge loss [47] as the GAN objective and apply spectral normalization [48]. We train all models using Adam optimizer [45] with a learning rate of 0.0002, $\beta_1 = 0.5, \beta_2 = 0.999$, and a batch size of of 64.

[4] `https://github.com/POSTECH-CVLab/PyTorch-StudioGAN`

The generator is trained for $100k$ iterations, and five updates of the discriminator are performed for every update of the generator. We select the model achieving best FID on the validation dataset.

Classification networks used for S2M sampling are obtained from multiple branches of MobileNet V2 architecture. We first train the classifier with only $\mathcal{L}_r$ during 200 epochs for CIFAR-7to3 and 30 epochs for CelebA-BMS, CelebA-ABMNSW dataset. We use SGD optimizer with a learning rate of 0.1 and cosine annealing for this training. Then, the classifier is trained with the sum of all classification losses for $30k$ iterations. For CIFAR-7to3 and CelebA-BMS dataset, we set the temperature of D_r and D_f as 0.2, 1.0, and 1.2 when the size of difference index set is 0, 1, and 2, respectively. For CelebA-ABMNSW dataset, we set the temperature of D_r and D_f as 0.1, 0.1, 1.0, 1.0, 1.6 and 1.6 when the size of difference index set is 0, 1, 2, 3, 4, 5 and 6, respectively. Each γ_k corresponding to the intersection set is set to 0.1 for CelebA-BMS and 0.5 for CelebA-ABMNSW. On CIFAR-7to3 dataset, we set 0.5 and 0.8 to this parameter for unconditional GAN and cGAN-PD, respectively.

D.3 CelebA-HQ

For CelebA-HQ dataset, we follow the original partition description and use the images with the resolution of 256×256. To make the training dataset annotated by single positive labels (*i.e.*, Black hair, Man and Smiling), we distribute the images belong to the joint classes equally to each corresponding classes of the single positive labels.

We use the pretrained StyleGAN V2 [49]. Classification networks used for S2M sampling are obtained from multiple branches of MobileNet V2 [46] architecture. We first train the classifier with only $\mathcal{L}_r$ during 30 epochs. We use SGD optimizer with a learning rate of 0.1 and cosine annealing for this training. Then, the classifier is trained with the sum of all classification losses for $1k$ iterations. We set the temperature of D_r as 0.2, 1.0, and 1.2 when the size of difference index set is 0, 1, and 2, respectively. Each γ_k corresponding to the intersection set is set to 0.5.

E Additional Experimental Results

E.1 2×16 Gaussians Example

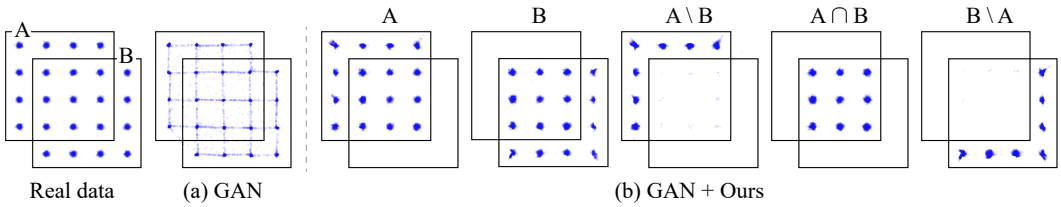


Figure 11: Example of 2×16 Gaussians. Using the outputs of original GAN, S2M Sampling samples high-quality points within various conditions $(\mathrm{A}, \mathrm{B}, \mathrm{A} \setminus \mathrm{B}, \mathrm{B} \setminus \mathrm{A}, \mathrm{A} \cap \mathrm{B})$.

To provide an illustrative example, we modify *25 Gaussians* [20, 21, 25] to have two 4×4 grids of two-dimensional Gaussians of two classes A and B (See Figure 11). The 23 modes in 2×16 Gaussians are horizontally and vertically spaced by 1.0 and have a standard deviation of 0.05. We first train a GAN to randomly draw points within two grids using WGAN-GP [39] as the GAN objective. To apply S2M sampling, we train the classifier to classify each point into two classes A and B. The generator, discriminator, and classification networks consist of ReLU activations and fully connected layers of input size: 2-512-512-512.

In this setting, S2M sampling attempts to draw points of given classes (A, B), the overlapping class ($A \cap B$), and the non-overlapping classes ($A \setminus B$, $B \setminus A$). We obtain samples at 400 MCMC iterations. As shown in Figure 11, S2M sampling correctly draw points of the target classes. On the other hand, GAN tends to generate spurious lines between the points. For the quantitative analysis, we report the accuracy, high-quality ratio, and mode standard deviation. We generate $10k$ samples and assign each point to the mode with the closest L_2 distance for measuring the accuracy. Following Turner *et al.* [21], samples whose L_2 distances are within four standard deviations are considered as

Table 5: Accuracy (%), high-quality ratio (%), and mode standard deviation on 2×16 Gaussians.

Condition	GAN			**GAN + Ours**		
	Accuracy	High Quality	Std. Dev.	Accuracy	High Quality	Std. Dev.
A, B	69.83$\pm$0.35	84.39$\pm$0.60	0.106$\pm$0.002	100.00$\pm$0.00	98.94$\pm$0.40	0.052$\pm$0.002
$\mathrm{A} \setminus \mathrm{B}, \mathrm{B} \setminus \mathrm{A}$	30.17$\pm$0.35	88.87$\pm$0.50	0.090$\pm$0.002	99.52$\pm$0.36	98.67$\pm$0.51	0.051$\pm$0.002
$\mathrm{A} \cap \mathrm{B}$	39.66$\pm$0.46	80.98$\pm$0.82	0.118$\pm$0.003	100.00$\pm$0.00	99.73$\pm$0.14	0.050$\pm$0.001

high-quality samples. As shown in Table 5, S2M sampling accurately draw samples for all conditions, and the ratio of high-quality samples is improved by 14.36% on average.

E.2 CelebA-HQ 256×256

To validate whether S2M sampling can be adopted to state-of-the-art architecture on high resolution (256×256) image dataset, we evaluate the performance of S2M sampling with the pretrained StyleGANv2 [49] on CelebA-HQ dataset. The quantitative results for StyleGANv2 with and without S2M sampling are shown in Table 6, and the qualitative results are depicted in Section E.5.

Table 6: Quantitative results of StyleGANv2 with S2M sampling.

Method	Accuracy (↑)	FID (↓)
StyleGANv2	15.44%	17.08
StyleGANv2 + **Ours**	77.18%	14.64

E.3 Data Embedding Visualization.

To visually probe the effect of S2M sampling, we perform t-SNE [50, 51] on generated samples. We embed Inception-V3 activations of samples generated by various models on CIFAR-7to3. As shown in Figure 12, S2M sampling accurately draw samples for all classes compared to other generative models.

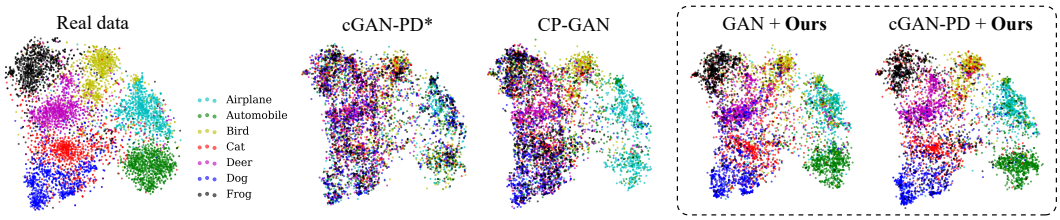


Figure 12: t-SNE visualization results for generated samples on CIFAR-7to3 dataset. Samples drawn from S2M sampling are embedded similarly to the real data.

E.4 Ablation Study

Since classifiers used in S2M sampling are not optimal in practice, we correct the sampling algorithm by scaling the temperatures of the classifiers and adjusting γ_k. Temperature scaling [30] is useful technique to control the confidence of classifier. If the temperature of D_f and D_r is small, samples that are likely to belong to overlapping classes tend to be drawn mostly in S2M sampling. Another approach is to adjust γ_k. We can obtain samples clearly distinct from classes of a difference set by decreasing γ_i for all $i \in I$.

To validate the effects of the temperature scaling and γ adjustment in S2M sampling, we perform the ablation studies on CIFAR-7to3 and CelebA-BMS dataset. The hyperparameter values used are explained in Appendix D. In Table 7, we report the average accuracy and FID when S2M sampling is applied to unconditional GAN and cGAN-PD. With the proper adjustment of the hyperparameters, accuracy can be highly improved while FID is maintained or slightly degraded.

Table 7: Ablation study for the hyperparameters of S2M sampling.

Method	Metric	CIFAR-7to3		CelebA-BMS	
		GAN	cGAN-PD	GAN	cGAN-PD
Sampling w/ actual logits	Acc. (↑)	62.73±1.63	72.33±0.52	71.14±1.46	73.32±4.62
	FID (↓)	14.99±0.17	14.11±0.23	9.86±1.41	10.29±0.83
+ temperature scaling	Acc. (↑)	68.40±0.82	76.32±1.44	74.99±1.95	78.50±2.63
	FID (↓)	14.81±0.23	14.12±0.31	9.81±1.21	10.21±0.41
+ γ adjustment	Acc. (↑)	77.65±1.22	80.62±2.08	85.22±4.27	90.44±1.05
	FID (↓)	14.42±0.55	14.14±0.34	10.50±0.97	10.63±0.29

E.5 Qualitative Results

In this section, we provide the qualitative results for the experiments on CIFAR-7to3, CelebA-BMS, and CelebA-HQ. The qualitative results for CIFAR-7to3 and CelebA-BMS are shown in Figure 13 and Figure 14, respectively. One of advantages of S2M sampling is that it can be readily built upon existing state-of-the-art GANs. Figure 15 represents the samples with the resolution of 256×256 when we adopted S2M sampling to pretrained StyleGAN V2 [49] on CelebA-HQ dataset.

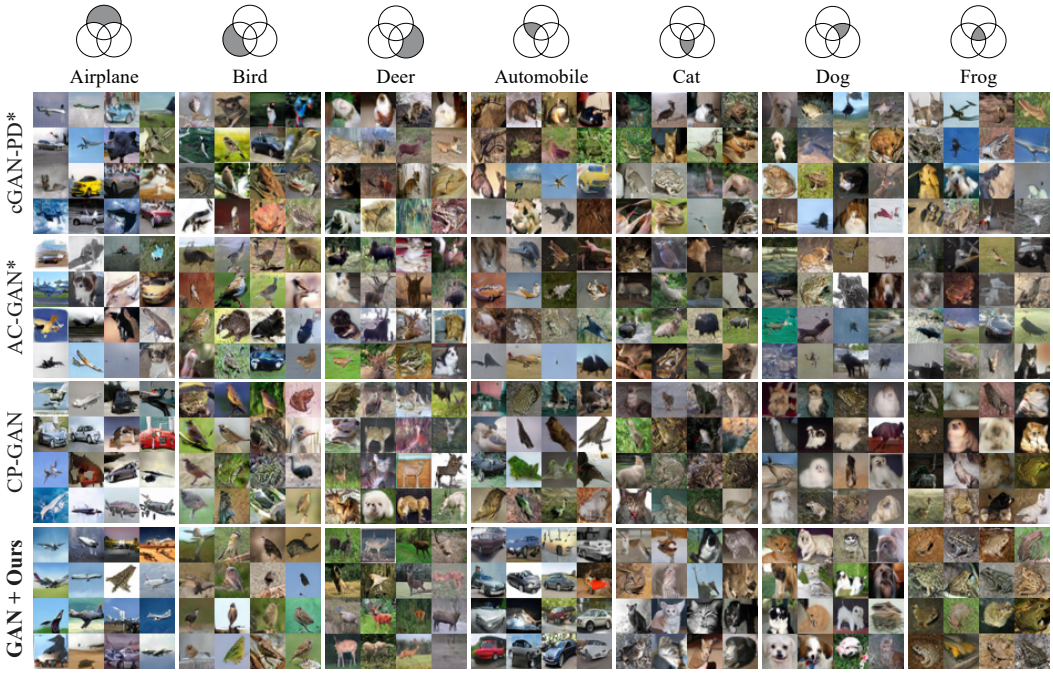


Figure 13: Qualitative results for cGAN-PD*, AC-GAN*, CP-GAN, and S2M sampling with GAN on CIFAR-7to3. For cGAN-PD*, AC-GAN* and CP-GAN, we provide $1/n$ as the labels for each class to generate images belonging to n classes.

F S2M Sampling with Different Attributes

In this section, we examine S2M sampling with various attributes appearing in CelebA dataset. Concretely, we conduct two additional datasets: (i) CelebA-BBM consisting of classes of *Brown-hair*, *Bushy-eyebrows*, and *Mouth-slightly-opens* attributes, (ii) CelebA-HBW consisting of classes of *High-cheekbones*, *Bags-under-eye*, and *Wavy-hair* attributes. Those attributes are chosen due to their strong visual impact. For both datasets, we adopt S2M sampling to pretrained StyleGANv2.

Quantitative Results. Table 8 shows the quantitative results of S2M sampling. As shown in Table 8, we can observe that S2M sampling correctly draws samples corresponding to each joint class for

Figure 14: Qualitative results for cGAN-PD*, AC-GAN*, CP-GAN, and S2M sampling with GAN on CelebA-BMS. For cGAN-PD*, AC-GAN* and CP-GAN, we provide $1/n$ as the labels for each class to generate images belonging to n classes. Intersections and differences are denoted by plus signs and minus signs, respectively.

Table 8: Quantitative results of StyleGANv2 with S2M sampling on two variants of CelebA-BMS: CelebA-BBM and CelebA-HBW.

Dataset	Method	Accuracy (↑)	FID (↓)
CelebA-BBM	StyleGANv2	18.41%	15.57
	StyleGANv2 + **Ours**	**74.97%**	**14.41**
CelebA-HBW	StyleGANv2	12.68%	15.80
	StyleGANv2 + **Ours**	**70.94%**	**14.54**

both datasets, which indicates that S2M sampling can be efficiently performed with various types of attributes.

Qualitative Results. For the qualitative results, we depict samples drawn by S2M sampling for three joint classes of CelebA-BBM and CelebA-HBW in Figure 16. For all joint classes, the samples in Figure 16 are randomly selected from the outputs of S2M sampling.

G Consideration of other sampling approaches for applying S2M Sampling

In this section, we provide insight to apply S2M sampling with other sampling methods. We mainly considered applying three sampling algorithms which can be found in previous GAN sampling studies [20, 21, 25, 29]; Rejection sampling, Independent Metropolis-Hastings algorithm, and Langevin dynamics. Here, we briefly discuss the pros and cons of each sampling method we faced in our problem settings.

Rejection Sampling As discussed in DRS [20], the rejection sampling can be applied to sample from the target distribution $p_t(x)$ if we can compute the ratio between the target density $p_t(x)$ and the generator density $p_G(x)$, and the upper bound of the density ratio $p_t(x)/p_G(x)$. However, it is in general difficult and expensive to compute this upper bound in the high dimensional data space.

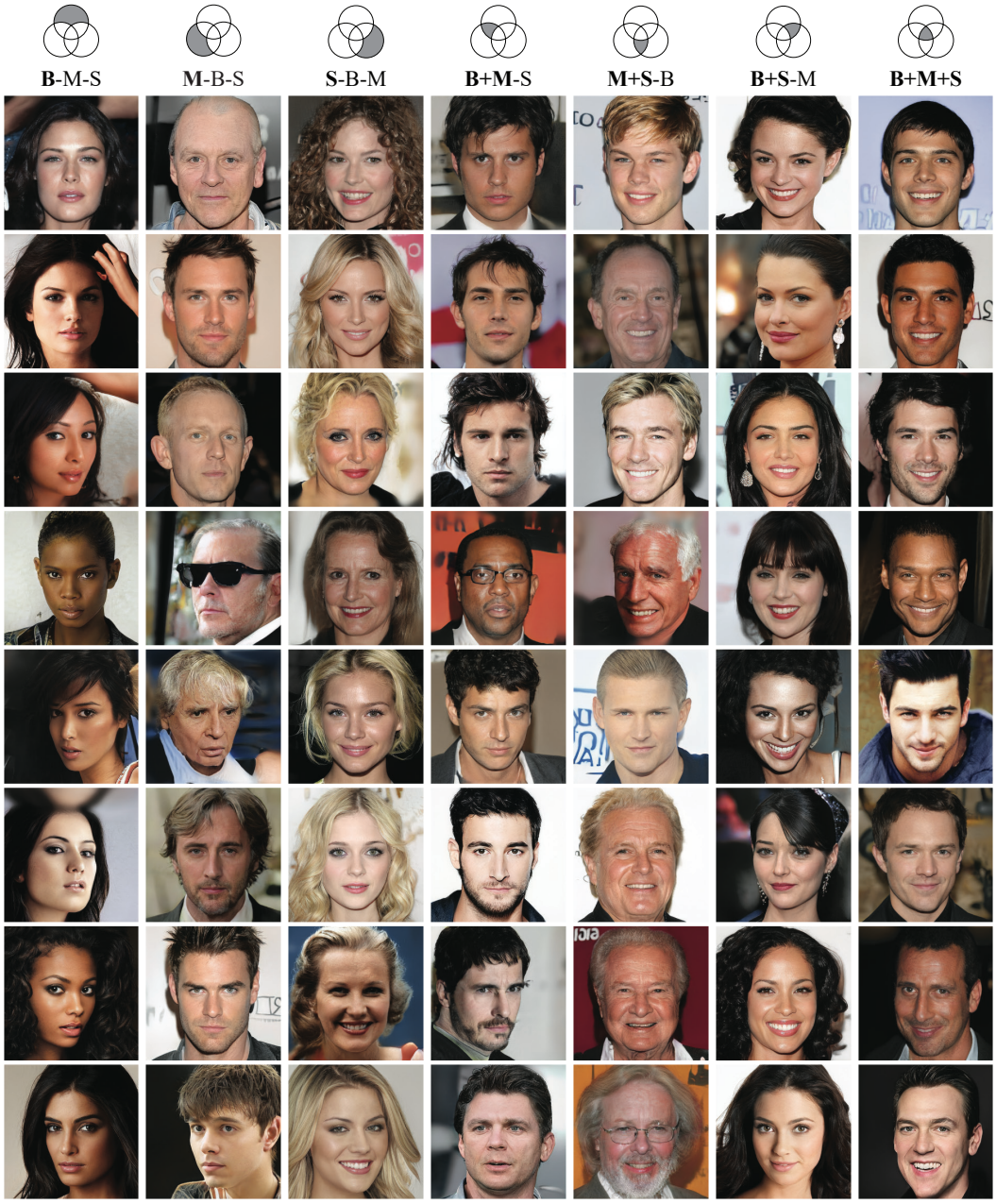


Figure 15: Qualitative results of applying S2M sampling to StyleGANv2 on CelebA-HQ. Intersections and differences are denoted by plus signs and minus signs, respectively.

We need several heuristics to mitigate the issues, e.g., shifting the logit score of the classifier used to compute the density ratio as introduced by DRS [20], which may introduce additional non-trivial efforts for hyperparameter searching.

Independent Metropolis-Hastings algorithm This algorithm can also be used to draw samples from the target distribution if we can compute the density ratio $p_t(x)/p_G(x)$. Unlike Rejection sampling, we do not need to compute the upper bound of this density ratio, but a sequence of samples forming a Markov chain is required. Our study mainly used this algorithm because it empirically performed well without complex heuristics and the sampling accuracy can be readily controlled at the cost of MCMC steps. To further mitigate the sample efficiency in our problem, we suggest the latent adaptation technique.

Figure 16: Qualitative results of applying S2M sampling to StyleGANv2 on two varaints of CelebA-BMS: CelebA-BBM (left) and CelebA-HBW (right).

Langevin dynamics Langevin dynamics is a gradient-based MCMC approach which can also be used when we can compute the density ratio $p_t(x)/p_G(x)$. Several studies [25, 29] employ its Euler-Maruyama discretization to improve the quality of GAN samples. While this algorithm can efficiently push a chain of samples towards the target distribution, the step size of the algorithm is very sensitive to the sampling cost and quality. Especially, in our problem, we need to deal with the case that the sample falls within the space where the gradient is not well-defined, i.e. $\operatorname{supp} p_G \setminus \operatorname{supp} p_t$. We did not use the algorithm as we could not find an effective way to address these issues.

H Potential Societal Impacts

This work demonstrates that it is possible to generate multi-label data from limited labels. In addition, this work can be freely adopted to unconditional GANs trained with a large amount of unlabeled

data. Hence, our work can reduce the high annotation cost that research groups face in common. Despite the fact that deep learning models tend to struggle from learning underrepresented data [52], properly calibrated sampling algorithm does not readily ignore rarely appearing data, meaning that it is unlikely to introduce bias into generative models.