Rethinking Generative Mode Coverage: A Pointwise Guaranteed Approach

Peilin Zhong* Yuchen Mo* Chang Xiao* Pengyu Chen Changxi Zheng

Columbia University (*Equal Contribution)

Missing Modes in Generative Models

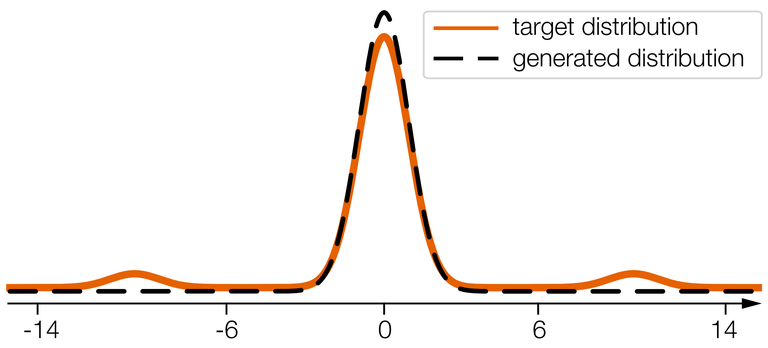

Figure 1: **Missing Modes Problem.**

- Motivating problem: Fitting a 1D target distribution P which has three modes, with one major mode and two minor mode.
- Conventional way: Reducing through training a statistical distance (such as f-divergence) between the generated distribution and provided data distribution.
- Issue: The statistical distance measures a **global** similarity between two distributions. A small global similarity does not imply a plausible mode coverage.
- Intuition: Optimizing a **local** similarity instead of global one.

Definition: Pointwise Coverage

We introduce an explicit notion of complete mode coverage, by switching from the global statistical distance to **local pointwise coverage**:
Provided a target data distribution P with a probability density $p(x)$ at each point x of the data space $\mathcal{X}$, we claim that a generator G has a complete mode coverage of P if the generator's probability $g(x)$ for generating x is pointwise lower bounded, that is

$$g(x) \leq \phi \cdot p(x), \forall x \in \mathcal{X},$$

where $\phi \in (0, 1)$ is a relaxation constant.

Properties of this metric:

- **Every point** x in the data space $\mathcal{X}$ will be generated by G with a finite and lower-bounded probability $g(x)$.
- **No** mode will be missed.
- Our metric implies the total variation distance between P and G is close (upper bounded by $1 - \phi$).

A Game-Theoretic Perspective

We offer an intuitive view of *why our mode coverage notion is attainable* through a game-theoretic lens.

- Consider a two-player game between Alice and Bob:
  - A target data distribution P and a family $\mathcal{G}$ of generators are given.
  - Alice chooses a generator $G \in \mathcal{G}$, and Bob chooses a data point $x \in \mathcal{X}$.
  - If $g(x) \geq \frac{1}{4}p(x)$, the game value is $v(G, x) = 1$, otherwise it is $v(G, x) = 0$.
  - Alice's goal is to maximize the game value, while Bob's goal is to minimize the game value.
- Consider two situations:
  - Bob first chooses a mixed strategy, a distribution Q over $\mathcal{X}$. Then, Alice chooses the best generator $G \in \mathcal{G}$. The expected game value is $\max_{G \in \mathcal{G}} \mathop{\mathbf{E}}_{x \sim Q}[v(G, x)]$.
  - Alice first chooses a mixed strategy, a distribution $R_{\mathcal{G}}$ of generators over $\mathcal{G}$. Then, Bob chooses $x \in \mathcal{X}$. The expected game value is $\min_{x \in \mathcal{X}} \mathbf{E}_{G \sim R_{\mathcal{G}}}[v(G, x)]$.
- Suppose $\forall Q$ over $\mathcal{X}$, $\exists G \in \mathcal{G}$ s.t. $\frac{1}{2}\int_{\mathcal{X}} |q(x) - g(x)| \, \mathrm{d}x \leq 0.1$, we can show:

$$\min_{Q} \max_{G \in \mathcal{G}} \mathop{\mathbf{E}}_{x \sim Q}[v(G, x)] \geq 0.4.$$

- Thus, $\max_{R_{\mathcal{G}}} \min_{x \in \mathcal{X}} \mathop{\mathbf{E}}_{G \sim R_{\mathcal{G}}}[v(G, x)] \geq 0.4$.
- $\forall x \in \mathcal{X}, g^*(x) \geq 0.1p(x)$.

Algorithm

Algorithm 1 Constructing a mixture of generators

1: **Parameters:** T, a positive integer number of generators, and $\delta \in (0, 1)$, a covering threshold.
2: **Input:** a target distribution P on a data domain $\mathcal{X}$.
3: For each $x \in \mathcal{X}$, initialize its weight $w_1(x) = p(x)$.
4: **for** $t = 1 \to T$ **do**
5: Construct a distribution P_t over $\mathcal{X}$ as follows:
6: For every $x \in \mathcal{X}$, normalize the probability density $p_t(x) = \frac{w_t(x)}{W_t}$, where $W_t = \int_{\mathcal{X}} w_t(x) \mathrm{d}x$.
7: Train a generative model G_t on the distribution P_t.
8: Estimate generated density $g_t(x)$ for every $x \in \mathcal{X}$.
9: For each $x \in \mathcal{X}$, if $g_t(x) < \delta \cdot p(x)$, set $w_{t+1}(x) = 2 \cdot w_t(x)$. Otherwise, set $w_{t+1}(x) = w_t(x)$.
10: **end for**
11: **Output:** a mixture of generators $\mathbf{G} = \{G_1, \ldots, G_T\}$.

Experiments

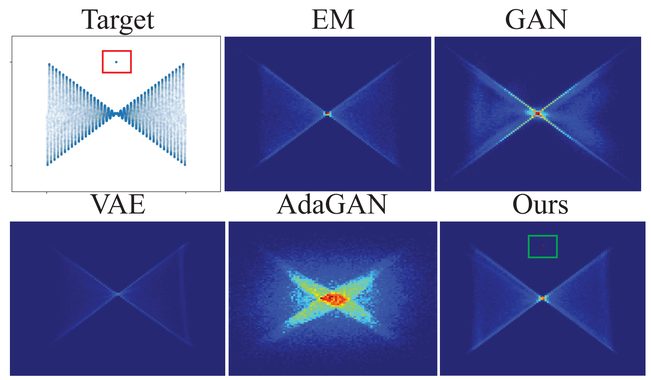

Figure 2: **Generative models on synthetic dataset**. The dataset consists of two modes: one major mode as an expanding sine curve ($y = x \sin 4x\pi$) and a minor mode as a Gaussian located at (10, 0)(highlighted in the reb box). We show color-coded distributions of generated samples from EM, GAN, VAE, AdaGAN, and our method. Only our method is able to cover the second mode.

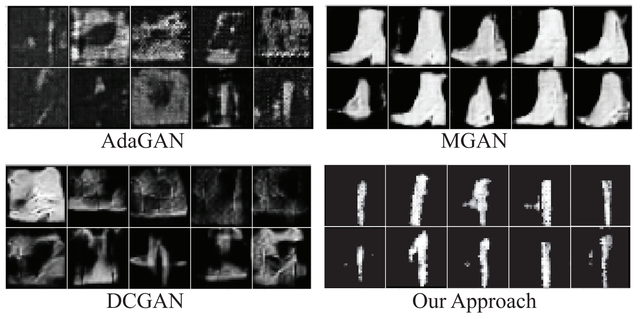

Figure 3: This dataset consists of the entire training dataset of FashionMNIST (with 60k images) mixed with randomly sampled 100 MNIST images labeled as "1". Then We train different generative models on this dataset. To evaluate this, we train an 11-class classifier to distinguish the 10 classes in Fashion-MNIST and one class in MNIST (i.e., "1"). Here we show samples that are generated by each tested methods and also classified by the pre-trained classifier most confidently as "1" images. Samples of our method are visually much closer to "1".

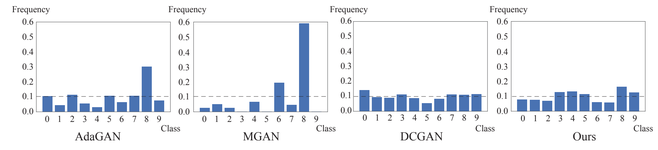

Figure 4: Training samples are drawn uniformly from each class. But generated samples by AdaGAN and MGAN are considerably nonuniform, while those from DCGAN and our method are more uniform. This experiment suggests that the conventional heuristic of reducing a statistical distance might not merit its use in training generative models.