CURRICULUM LEARNING FOR DEEP GENERATIVE MODELS WITH CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Training generative models like Generative Adversarial Network (GAN) is challenging for noisy data. A novel curriculum learning algorithm pertaining to clustering is proposed to address this issue in this paper. The curriculum construction is based on the centrality of underlying clusters in data points. The data points of high centrality takes priority of being fed into generative models during training. To make our algorithm scalable to large-scale data, the active set is devised, in the sense that every round of training proceeds only on an active subset containing a small fraction of already trained data and the incremental data of lower centrality. Moreover, the geometric analysis is presented to interpret the necessity of cluster curriculum for generative models. The experiments on cat and human-face data validate that our algorithm is able to learn the optimal generative models (e.g. ProGAN) with respect to specified quality metrics for noisy data. An interesting finding is that the optimal cluster curriculum is closely related to the critical point of the geometric percolation process formulated in the paper.

1 INTRODUCTION

Deep generative models pique researchers' interest in the past decade. The fruitful progress has been achieved on this topic, such as auto-encoder (Hinton & Salakhutdinov, 2006) and variational auto-encoder (VAE) (Kingma & Welling, 2013; Rezende et al., 2014), generative adversarial network (GAN) (Goodfellow et al., 2014; Radford et al., 2016; Arjovsky et al., 2017), normalizing flow (Rezende & Mohamed, 2015; Dinh et al., 2015; 2017; Kingma & Dhariwal, 2018), and auto-regressive models (van den Oord et al., 2016b;a; 2017). However, it is non-trial to train a deep generative model that can converge to a proper minimum of associated optimization. For example, GAN suffers non-stability, model collapse, and generative distortion during training. Many insightful algorithms have been proposed to circumvent those issues, including feature engineering (Salimans et al., 2016), various discrimination metrics (Mao et al., 2016; Arjovsky et al., 2017; Berthelot et al., 2017), distinctive gradient penalties (Gulrajani et al., 2017; Mescheder et al., 2018), spectral normalization to discriminator (Miyato et al., 2018), and orthogonal regularization to generator (Brock et al., 2019). What is particularly of interest is that the breakthrough for GAN has been made with a simple technique of progressively growing neural networks of generator and discriminator from low-resolution images to high-resolution counterparts (Karras et al., 2018a). This kind of progressive growing also helps push the state of the art to a new level by enabling StyleGAN to produce photo-realistic and detail-sharp results (Karras et al., 2018b), shedding new light on wide applications of GANs in solving real problems. This idea of progressive learning is actually a general manner of cognition process (Elman, 1993; Oudeyer et al., 2007), which has been formally named curriculum learning in machine learning (Bengio et al., 2009). The central topic of this paper is to explore a new curriculum for training deep generative models.

To facilitate robust training of deep generative models with noisy data, we propose curriculum learning with clustering. The key contributions are listed as follows:

- We first summarize four representative curricula for generative models, i.e. architecture (generation capacity), semantics (data content), dimension (data space), and cluster (data distribution). Among these curricula, cluster curriculum is newly proposed in this paper.

- Cluster curriculum is to treat data according to centrality of each data point, which is pictorially illustrated and explained in detail. To foster large-scale learning, we devise the active set algorithm that only needs an active data subset of small fixed size for training.
- The geometric principle is formulated to analyze hardness of noisy data and advantage of cluster curriculum. The geometry pertains to counting a small sphere packed in an ellipsoid, on which is based the percolation theory we use.

The research on curriculum learning is diverse. Our work focuses on curricula that are closely related to data attributes, beyond which is not the scope we concern in this paper.

2 CURRICULUM LEARNING

Curriculum learning has been a basic learning approach to promoting performance of algorithms in machine learning. We quote the original words from the seminal paper (Bengio et al., 2009) as its definition:

Curriculum learning. *"The basic idea is to start small, learn easier aspects of the task or easier sub-tasks, and then gradually increase the difficulty level" according to pre-defined or self-learned curricula.*

From cognitive perspective, curriculum learning is common for human and animal learning when they interact with environments (Elman, 1993), which is the reason why it is natural as a learning rule for machine intelligence. The learning process of cognitive development is *gradual* and *progressive* (Oudeyer et al., 2007). In practice, the design of curricula is task-dependent and data-dependent. Here we summarize the representative curricula that are developed for generative models.

Architecture curriculum. The deep neural architecture itself can be viewed as a curriculum from the viewpoint of learning concepts (Hinton & Salakhutdinov, 2006; Bengio et al., 2006) or disentangling representations (Lee et al., 2011). For example, the different layers decompose distinctive features of objects for recognition (Lee et al., 2011; Zeiler & Fergus, 2014; Zhou et al., 2016) and generation (Bau et al., 2018). Besides, Progressive growing of neural architectures is successfully exploited in GANs (Karras et al., 2018a; Heljakka et al., 2018; Korkinof et al., 2018; Karras et al., 2018b).

Semantics curriculum. The most intuitive content for each datum is the semantic information that the datum conveys. The hardness of semantics determines the difficulty of learning knowledge from data. Therefore, the semantics can be a common curriculum. For instance, the environment for a game in deep reinforcement learning (Justesen et al., 2018) and the number sense of learning cognitive concepts with neural networks (Zou & McClelland, 2013) can be such curricula.

Dimension curriculum. The high dimension usually poses the difficulty of machine learning due to the curse of dimensionality (Donoho, 2000), in the sense that the amount of data points for learning grows exponentially with dimension of variables (Vershynin, 2018). Therefore, the algorithms are expected to be beneficial from growing dimensions. The effectiveness of dimension curriculum is evident from recent progress on deep generative models, such as ProGANs (Karras et al., 2018a;b) by gradually enlarging image resolution and language generation from short sequences to long sequences of more complexity (Rajeswar et al., 2017; Press et al., 2017).

3 CLUSTER CURRICULUM

For fitting distributions, dense data points are generally easier to handle than sparse data or outliers. To train generative models robustly, therefore, it is plausible to raise cluster curriculum, meaning that generative algorithms first learn from data points close to cluster centers and then with more data progressively approaching cluster boundaries. Thus the stream of feeding data points to models for curriculum learning is the process of clustering data points according to cluster centrality that will be explained in section 3.2. The toy example in Figure 1 illustrates how to form cluster curriculum.

3.1 WHY CLUSTERS MATTER

The importance of clusters for data points is actually obvious from geometric point of view. The data sparsity in high-dimensional spaces causes the difficulty of fitting the underlying distribution of

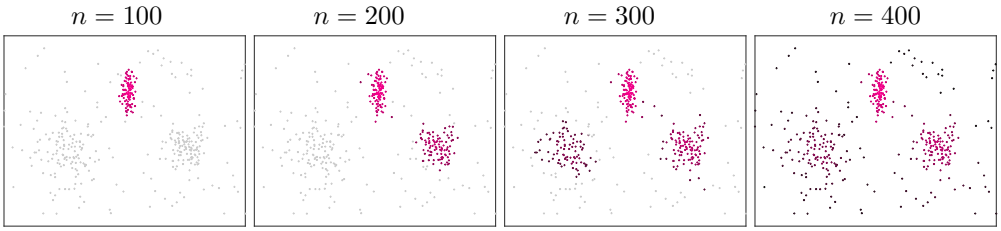


Figure 1: Cluster Curriculum. From magenta color to black color, the centrality of data points reduces. The value n is the number of data points taken with centrality order.

data points (Vershynin, 2018). So generative algorithms may be beneficial when proceeding from the local spaces where data points are relatively dense. Such data points form clusters that are generally informative subsets with respect to entire dataset. In addition, clusters contain common regular patterns of data points, where generative models are easier to converge. What is most important is that *noisy data points deteriorate performance of algorithms*. For classification, the effectiveness of curriculum learning is theoretically proven to circumvent the negative influence of noisy data (Gong et al., 2016). We will analyze this aspect for generative models with geometric facts.

3.2 Generative models with cluster curriculum

With cluster curriculum, we are allowed to gradually learn generative models from dense clusters to cluster boundaries and finally to all data points. In this way, generative algorithms are capable of avoiding the direct harm of noise or outliers. To this end, we first need a measure called centrality that is the terminology in graph-based clustering. It quantifies the compactness of a cluster in data points or a community in complex networks (Newman, 2010). A large centrality implies that the associated data point is close to one of cluster centers. For easy reference, we provide the algorithm of the centrality we use in appendix. For experiments in this paper, all the cluster curricula are constructed by the centrality of stationary probability distribution, i.e. the eigenvector corresponding to the largest eigenvalue of the transition probability matrix drawn from the data.

To be specific, let $\boldsymbol{c} \in \mathcal{R}^m$ denote the centrality vector of m data points. Namely, the i-th entry c_i of $\boldsymbol{c}$ is the centrality of data point $\boldsymbol{x}_i$. Sorting $\boldsymbol{c}$ in descending order and adjusting the order of original data points accordingly give data points arranged by cluster centrality. Let $\overrightarrow{\mathcal{X}} = \{\overrightarrow{\mathcal{X}}_0, \overrightarrow{\mathcal{X}}_1, \ldots, \overrightarrow{\mathcal{X}}_l\}$ signify the set of centrality-sorted data points, where $\overrightarrow{\mathcal{X}}_0$ is the base set that guarantees a proper convergence of generative models, and the rest of $\overrightarrow{\mathcal{X}}$ is evenly divided into l subsets according to centrality order. In general, the number of data points in $\overrightarrow{\mathcal{X}}_0$ is moderate compared to m and determined according to $\mathcal{X}$. Such division of $\overrightarrow{\mathcal{X}}_0$ serves to efficiency of training, because we do not need to train models from a very small dataset. The cluster curriculum learning is carried out by incrementally feeding subsets in $\overrightarrow{\mathcal{X}}$ into generative algorithms. In other words, algorithms are successively trained on $\overrightarrow{\mathcal{X}}_0 \leftarrow \overrightarrow{\mathcal{X}}_0 \cup \overrightarrow{\mathcal{X}}_i$ after $\overrightarrow{\mathcal{X}}_0$, meaning that the curriculum for each round of training is accumulated with $\overrightarrow{\mathcal{X}}_i$.

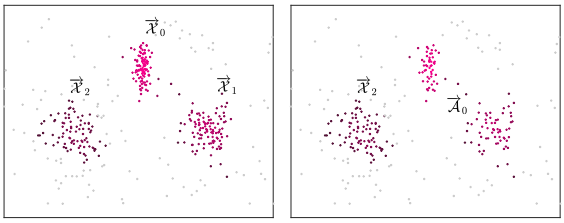
In order to determine the optimal curriculum $\overrightarrow{\mathcal{X}}_0 \cup \overrightarrow{\mathcal{X}}_1 \cup \cdots \cup \overrightarrow{\mathcal{X}}_i$, we need the aid of quality metric of generative models, such as Fréchet inception distance (FID) or sliced Wasserstein distance (SWD) (Borji, 2018). For generative models trained with each curriculum, we calculate the associated score s_i via the specified quality metric. The optimal curriculum for effective training can be identified by the minimal value for all s_i, where $i = 1, \ldots, l+1$. The interesting phenomenon of this score curve will be illustrated in the experiment. The minimum of score $\boldsymbol{s}$ is apparently metric-dependent. One can refer to (Borji, 2018) for the review of evaluation metrics. In practice, we can opt one of reliable metrics to use or multiple metrics for decision-making of the optimal model.

There are two ways of using incremental subset $\overrightarrow{\mathcal{X}}_i$ during training. One is that the parameters of models are re-randomized when new data are used, the procedure of which is given in Algorithm 1 in appendix. The other is that the parameters are fine-tuned based on pre-training of previous model, which will be presented with a fast learning algorithm in the following section.

3.3 ACTIVE SET FOR SCALABLE TRAINING

To obtain the precise minimum of $\boldsymbol{s}$, the cardinality of $\overrightarrow{\mathcal{X}}_i$ needs to be set much smaller than m, meaning that l will be large even for a dataset of moderate scale. The training of many loops will lead to time-consuming. Here we propose the active set to address the issue, in the sense that for each loop of cluster curriculum, generative models are always trained with a subset of small fixed size instead of $\overrightarrow{\mathcal{X}}_0 \leftarrow \overrightarrow{\mathcal{X}}_0 \cup \overrightarrow{\mathcal{X}}_i$ whose size becomes incrementally large. To form the active set $\mathcal{A}$, the subset $\overrightarrow{\mathcal{A}}_0$ of data points are randomly sampled from $\overrightarrow{\mathcal{X}}_0$ to combine with $\overrightarrow{\mathcal{X}}_i$ for the next loop, where $|\overrightarrow{\mathcal{A}}_0| = |\mathcal{A}| - |\overrightarrow{\mathcal{X}}_i|$. For easy understanding, we illustrate the active set with toy example in Figure 2. In this scenario, *progressive pre-training* must be applied, meaning that the update of model parameters for the current training is based on parameters of previous loop. The procedure of cluster curriculum with active set is detailed in Algorithm 2 in appendix.

The active set allows us to train generative models with a small dataset that is actively adapted, thereby significantly reducing the training time for large-scale data.



(a) $\overrightarrow{\mathcal{X}} = \{\overrightarrow{\mathcal{X}}_0, \overrightarrow{\mathcal{X}}_1, \overrightarrow{\mathcal{X}}_2\}$ (b) $\mathcal{A} = \overrightarrow{\mathcal{A}}_0 \cup \overrightarrow{\mathcal{X}}_2$

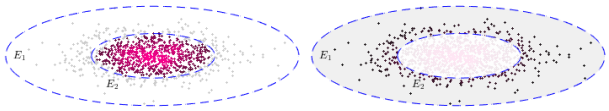
Figure 2: Schematic illustration of active set for cluster curriculum. Here $|\overrightarrow{\mathcal{X}}_0| = |\overrightarrow{\mathcal{X}}_1| = |\overrightarrow{\mathcal{X}}_2| = 100$. The cardinality $|\mathcal{A}|$ of the active set $\mathcal{A}$ is 200. When $\overrightarrow{\mathcal{X}}_2$ is taken for training, we need to randomly sample another 100 (i.e. $|\mathcal{A}| - |\overrightarrow{\mathcal{X}}_2|$) data points from the history data $\overrightarrow{\mathcal{X}}_0 \cup \overrightarrow{\mathcal{X}}_1$ to form $\overrightarrow{\mathcal{A}}_0$. Then the complete active set is composed by $\mathcal{A} = \overrightarrow{\mathcal{A}}_0 \cup \overrightarrow{\mathcal{X}}_2$. We can see that data points in $\overrightarrow{\mathcal{X}}_0 \cup \overrightarrow{\mathcal{X}}_1$ become less dense after sampling.

4 GEOMETRIC VIEW OF CLUSTER CURRICULUM

Cluster curriculum bears the interesting relation to high-dimensional geometry, which can provide geometric understanding of our algorithm. Without loss of generality, we work on a cluster obeying normal distribution. The characteristic of the cluster can be extended into other clusters of the same distribution. For easy analysis, let us begin with a toy example. As Figure 3(a) shows, the confidence ellipse E_2 fitted from the subset of centrality-ranked data points is nearly conformal to E_1 of all data points, which allow us to put the relation of these two ellipses by virtue of the confidence-level equation. Let $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ signify the center and covariance matrix of the cluster $\mathcal{C}$ of interest, where $\mathcal{C} = \{\boldsymbol{x}_i | \boldsymbol{x}_i \in \mathbb{R}^d, i = 1, \ldots, n\}$. To make it formal, we can write the equation by

$$\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x} = \chi^2_{d,\alpha}, \tag{1}$$

where $\chi^2_{d,\alpha}$ can be the chi-squared distribution or Mahalanobis distance square, d is the degree of freedom, and $(1-\alpha) \in [0, 1]$ is the confidence level. For conciseness, we write $\chi^2_{d,\alpha}$ as χ^2_α in the following context. Then the ellipses E_{α_1} and E_{α_2} correspond to $\chi^2_{\alpha_1}$ and $\chi^2_{\alpha_2}$, respectively, where $\alpha_1 \leq \alpha_2$.



(a) (b)

Figure 3: Illustration of growing one cluster in cluster curriculum. (a) Data points taken with large centrality. (b) The annulus formed by of removing the inner ellipse from the outer one.

To analyze the hardness of training generative model, a fundamental aspect is to *examine the number $n(E)$ of given data points falling in a geometric entity E* [1] *and the number $N(E)$ of lattice points in it*. The less $n(E)$ is compared to $N(E)$, the harder the problem will be. However, the enumeration of lattice points is computationally prohibitive for high dimensions. Inspired by the information theory of encoding data of normal distributions (Roman, 1996), we count the number of small spheres S_ε of radius ε packed in the ellipsoid E instead. Thus we can use this number to replace the role of $N(E)$ as long as the radius of the sphere S_ε is set properly. With a little abuse of notation, we still use $N(E)$ to denote the packing number in the following context. Theorem 1 gives the exact form of $N(E)$.

[1] For cluster curriculum, it is an annulus explained shortly.

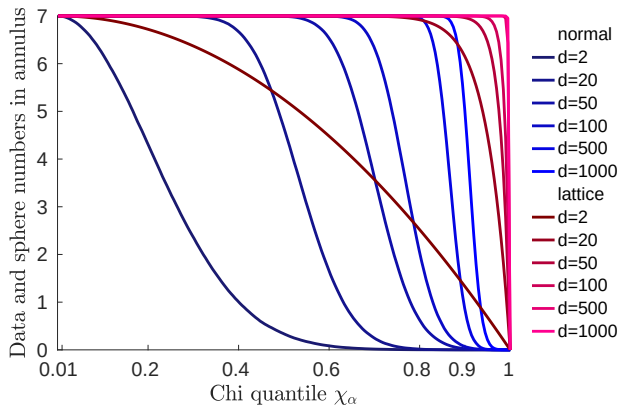


Figure 4: Comparison between the number $n(A)$ of data points sampled from isotropic normal distributions and $N(A)$ of spheres (lattice) packed in the annulus A with respect to the Chi quantile χ_{α_2}. d is the dimension of data points. For each dimension, we sample 70,000 data points from $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$. The scales of y-axis and x-axis are normalized by 10,000 and χ_{α_1}, respectively.

Theorem 1. *For a set $\mathcal{C} = \{\boldsymbol{x}_i | \boldsymbol{x}_i \in \mathbb{R}^d\}$ of n data points drawn from normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, the ellipsoid E_α of confidence $1-\alpha$ is defined as $\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x} \leq \chi_\alpha^2$, where $\boldsymbol{\Sigma}$ has no zero eigenvalues and $\alpha \in [0, 1]$. Let $N(E_\alpha)$ be the number of spheres of radius ε packed in the ellipsoid E_α. Then we can establish*

$$N(E_\alpha) = \left(\frac{\chi_\alpha}{\varepsilon}\right)^d \sqrt{\det(\boldsymbol{\Sigma})}. \tag{2}$$

We see that $N(E_\alpha)$ admits a tidy form with Mahalanobis distance χ_α, dimension d, and sphere radius ε as variables. The proof is provided in appendix.

The geometric region of interest for cluster curriculum is the annulus A formed by removing the ellipsoid [2] E_{α_2} from the ellipsoid E_{α_1}, as Figure 3(b) displays. We investigate the varying law between $n(A)$ and $N(A)$ in the annulus A when the inner ellipse E_{α_2} grows with cluster curriculum. For this purpose, we need the following two corollaries that immediately follows from Theorem 1.

Corollary 1. *Let $N(A)$ be the number of spheres of radius ε packed in the annulus A that is formed by removing the ellipsoid E_{α_1} from the ellipsoid E_{α_1}, where $\alpha_1 \leq \alpha_2$. Then the following identity holds*

$$N(A) = \left(\left(\frac{\chi_{\alpha_1}}{\varepsilon}\right)^d - \left(\frac{\chi_{\alpha_2}}{\varepsilon}\right)^d\right) \sqrt{\det(\Sigma)}. \tag{3}$$

Corollary 2. $N(A)/N(E_{\alpha_1}) = 1 - \left(\chi_{\alpha_2}/\chi_{\alpha_1}\right)^d$.

It is obvious that $N(A)$ goes infinite when $d \to \infty$ under the conditions that $\chi_{\alpha_1} > \chi_{\alpha_2}$ and ε is bounded. Besides, when E_{α_2} (cluster) grows, $N(A)$ reduces with exponent d if the ellipsoid E_{α_1} is fixed.

In light of Corollary 1, we can now demonstrate the functional law between $n(A)$ and $N(A)$. First, we determine χ_{α_1} as follows

$$\chi_{\alpha_1} = \max \|\boldsymbol{x}_i\|, \quad \boldsymbol{x}_i \in \mathcal{C}, \tag{4}$$

which means that E_{α_1} is the ellipsoid of minimal Mahalanobis distance to the center that contains all the data points in the cluster. In addition, we need to estimate a suitable sphere radius ε, such that $n(E_{\alpha_1})$ and $N(E_{\alpha_1})$ have comparable scales in order to make $n(A)$ and $N(A)$ comparable in scale. To achieve this, we define an oracle ellipse E where $n(E) = N(E)$. For simplicity, we let E_{α_1} be the oracle ellipse. Thus we can determine ε with Corollary 3.

Corollary 3. *If we let E_{α_1} be the oracle ellipse such that $n(E_{\alpha_1}) = N(E_{\alpha_1})$, then the free parameter ε can be computed with $\varepsilon = \chi_{\alpha_1}\left(\sqrt{\det(\boldsymbol{\Sigma})}/n(E_{\alpha_1})\right)^{\frac{1}{d}}$.*

To make the demonstration amenable to handle, data points we use for simulation are assumed to obey the isotropic normal distribution, meaning that data points are generated with nearly equal variance along each dimension. Figure 4 shows that $n(A)$ gradually exhibits the critical phenomena

[2] The ellipse refers to the surface and the ellipsoid refers to the elliptic ball.

Figure 5: Examples of LSUN cat dataset and CelebA face dataset. The samples in the first row are of high centrality and the samples of low centrality in the second row are noisy data or outliers that we call in the context.

of percolation processes[3] when the dimension d goes large, implying that the data points in the annulus A are significantly reduced when E_{α_2} grows a little bigger near the critical point. In contrast, the number $N(A)$ of lattice points is still large and varies negligibly until E_{α_2} approaches the boundary. This discrepancy indicates clearly that fitting data points in the annulus is pretty hard and guaranteeing the precision is nearly impossible when crossing the critical point of $n(A)$ even for a moderate dimension (e.g. $d = 500$). Therefore, the plausibility of cluster curriculum can be drawn naturally from this geometric fact.

5 EXPERIMENT

The generative model that we use for experiments are Progressive growing of GAN (ProGAN) (Karras et al., 2018a). This algorithm is chosen because ProGAN is the state-of-the-arts algorithm of GANs with official open sources available. According to convention, we opt the Fréchet inception distance (FID) (Borji, 2018) for ProGAN as the quality metric.

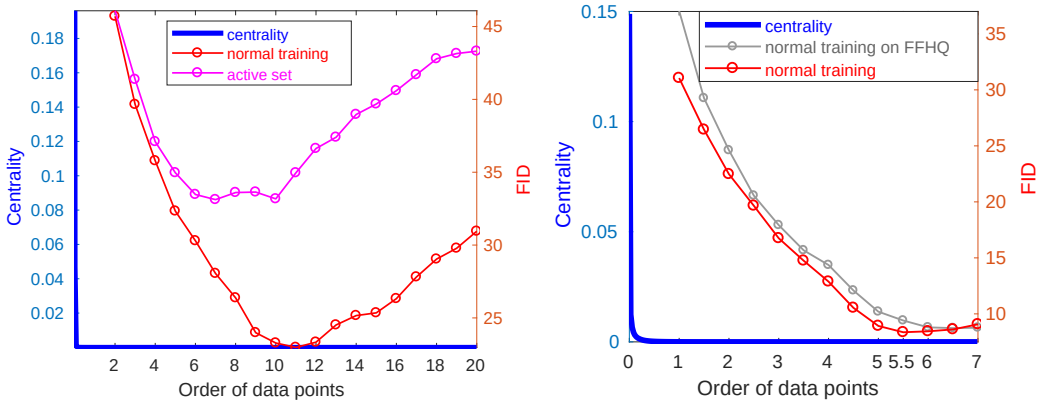
5.1 DATASET AND EXPERIMENTAL SETTING

We randomly sample 200,000 cat images from the LSUN dataset (Yu et al., 2015). These cat images are captured in the wild. So their styles vary significantly. Figure 5 shows the cat examples of high and low centralities. We can see that noisy cat images differ much from the clean ones. There actually contain the images of very few informative cat features, which are outliers we refer to. The curriculum parameters are set as $|\overrightarrow{\mathcal{X}}_0| = 20,000$ and $|\overrightarrow{\mathcal{X}}_i| = 10,000$, which means that the algorithms are trained with 20,000 images first and after the initial training, another 10,000 images according to centrality order are merged into the current training data for further re-training. For active set, its size is fixed to be $30,000$.

The CelebA dataset is a large-scale face attribute dataset (Liu et al., 2015). We use the cropped and well-aligned faces with a bit of image backgrounds preserved for generation task. For cluster-curriculum learning, we randomly sample 70,000 faces as the training set. The face examples of different centralities are shown in Figure 5. The curriculum parameters are set as $|\overrightarrow{\mathcal{X}}_0| = 10,000$ and $|\overrightarrow{\mathcal{X}}_i| = 5,000$. We bypass the experiment of active set on faces because it is used for large-scale data.

Each image in two databases is resized to be 64×64. To form cluster curricula, we exploit ResNet34 (He et al., 2016) pre-trained on ImageNet (Russakovsky et al., 2015) to extract 512-dimensional features for each face and cat images. The directed graphs are built with these feature vectors. We determine the parameter σ of edge weights by enforcing the geometric mean of weights to be 0.8. The number of nearest neighbors is set to be $K = 4 \log m$. The centrality is the stationary probability distribution. All codes are written with TensorFlow.

[3]Percolation theory is a fundamental tool of studying the structure of complex systems in statistical physics and mathematics. The critical point is the percolation threshold where the transition takes place. One can refer to (Stauffer & Aharony, 1994) if interested.



(a) Cat (b) Face

Figure 6: FID curves of cluster-curriculum learning for ProGAN on cat dataset and CelebA face dataset. The centrality and the FID share the x-axis due to that they have the same order of data points. The same colors of the y-axis labels and the curves denote the figurative correspondence. The network parameters for "normal training" are randomly re-initialized for each re-training. The active set is based on progressive pre-training of fixed small dataset. The scale of the x-axis is normalized by 10,000.

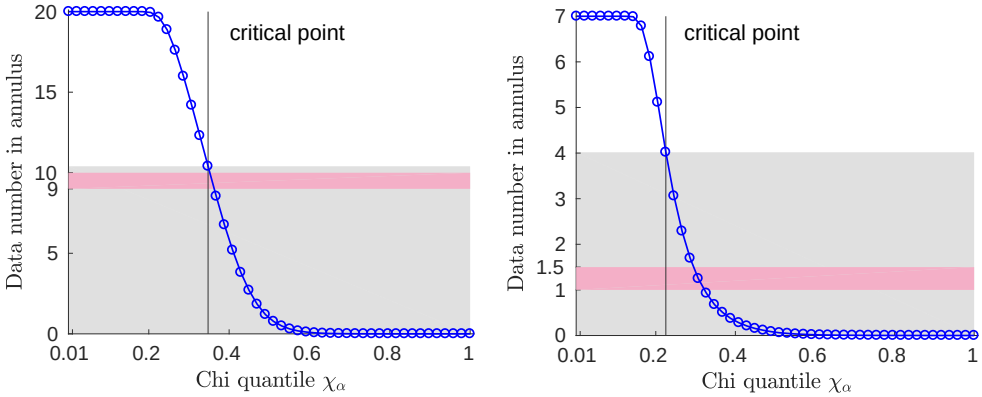
5.2 Experimental Result

From Figure 6a, we can see that the FID curves are all nearly V-shaped, indicating that global minima exist amid the training process. This is clear evidence that noisy data and outliers deteriorate the quality of generative models during training. From the optimal curricula found by two algorithms (i.e. curricula at 110,000 and 100,000), we can see that the curriculum of the active set differs from that of normal training only by one-step data increment, implying that the active set is reliable for fast cluster-curriculum learning. The performance of the active set measured by FID is much worse than that of normal training, especially when more noisy data are fed into generative models. However, this does not change the whole V-shape of the accuracy curve. Namely, it is applicable as long as the active set admits the metric minimum corresponding to the appropriate curriculum.

The V-shape of the centrality-FID curve on the cat data is due to that the noisy data of low centrality contains little effective information to characterize the cats, as already displayed in Figure 5. However, it is different for the CelebA face dataset where the face images of low centrality also convey part of face features. As evident by Figure 6b, ProGAN keeps being optimized by the majority of data until the curriculum of size $55,000$. To highlight the meaning of this nearly negligible minimum, we also conduct the exactly same experiment on the FFHQ face dataset containing $70,000$ face images of high-quality (Karras et al., 2018b). For FFHQ data, the noisy face data can be ignored. The gray curve of normal training in Figure 6b indicates that the FID of ProGAN is monotonically decreased for all curricula. This gentle difference of the FID curves at the ends between CelebA and FFHQ clearly demonstrate the difficulty of noisy data to generative algorithms.

5.3 Geometric Investigation

To understand cluster curriculum deeply, we employ the geometric method formulated in section 4 to analyze the cat and face data. The percolation processes are both conducted with 512-dimensional features from ResNet34. Figure 7 displays the curve of $n(A)$ that is the variable of interest in this scenario. As expected, the critical point in the percolation process occurs for both cases, as shown by blue curves. An obvious fact is that the optimal curricula (red strips) both fall into the (feasible) domains of percolation processes after critical points, as indicated by gray color. This is a desirable property because data become rather sparse in the annuli when crossing the critical points. Then noisy data play the non-negligible role in tuning parameters of generative models. Therefore, a fast learning strategy can be derived from percolation process. The training may begin from the curriculum specified by the critical point, thus significantly accelerating cluster-curriculum learning.



(a) cat data (b) face data

Figure 7: Geometric phenomenon of cluster curriculum on LSUN cat and CelebA face datasets. The pink strips are intervals of optimal curricula derived by generative models. For example, the value 9 of the pink interval in (a) is obtained by $9 = 20 - 11$, where 11 is one of the minima (i.e. 110,000) in Figure 6a. The others are derived in the same way. The subtraction transforms the data number in the cluster to be the one in the annulus. The critical points are determined by searching the maxima of the absolute discrete difference of the associated curves. The scales of y-axes are normalized by 10,000.

Another intriguing phenomenon is that the more noisy the data, the closer the optimal interval (red strip) is to the critical point. We can see that the optimal interval of the cat data is much closer to the critical point than that of the face data. What surprises us here is that the optimal interval of cluster curricula associated with the cat data nearly coincides with the critical point of the percolation process in the annulus! This means that the optimal curriculum may be found at the intervals close to the critical point of of $n(A)$ percolation for heavily noisy data, thus affording great convenience to learning an appropriate generative model for such datasets.

6 Analysis and Conclusion

The cluster curriculum is proposed for robust training of generative models. The active set of cluster curriculum is devised to facilitate scalable learning. The geometric principle behind cluster curriculum is analyzed in detail as well. The experimental results on LSUN cat dataset and CelebA face dataset demonstrate that generative models trained with cluster curriculum is capable of learning the optimal parameters with respect to the specified quality metric such as Fréchet inception distance and sliced Wasserstein distance. Geometric analysis indicates that the optimal curricula obtained from generative models are closely related to critical points of the associated percolation processes established in this paper. This intriguing geometric phenomenon is worth being explored deeply in terms of the theoretical connection between generative models and high-dimensional geometry.

It is worth emphasizing that the meaning of model optimality refers to the global minimum of the centrality-FID curve. As we already noted, the optimality is metric-dependent. We are able to obtain the optimal model with cluster curriculum, which does not mean that the algorithm only serves to this purpose. We know that more informative data can help learn a more powerful model covering large data diversity. Here a trade-off arises, i.e. the robustness against noise and the capacity of fitting more data. The centrality-FID curve provides a visual tool to monitor the state of model training, thus aiding us in understanding the learning process and selecting suitable models according to noisy degree of given data. For instance, we can pick the trained model close to the optimal curriculum for heavily noisy data or the one near the end of the centrality-FID curve for datasets of little noise. In fact, this may be the most common way of using cluster curriculum.

In this paper, we do not investigate the cluster-curriculum learning for the multi-class case, e.g. the ImageNet dataset with BigGAN (Brock et al., 2019). The cluster-curriculum learning of multiple classes is more complex than that we have already analyzed on face and cat data. We leave this study for future work.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. In *arXiv:1701.07875*, 2017.

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. GAN dissection: Visualizing and understanding generative adversarial networks. *arXiv:1811.10597*, 2018.

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2006.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning (ICML)*, pp. 41–48, 2009.

David Berthelot, Thomas Schumm, and Luke Metz. BEGAN: boundary equilibrium generative adversarial networks. *arXiv:1703.10717*, 2017.

Ali Borji. Pros and cons of GAN evaluation measures. *arXiv:1802.03446*, 2018.

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

Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation. In *International Conference on Learning Representations (ICLR)*, 2015.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations (ICLR)*, 2017.

David L. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. In *AMS Math Challenges Lecture*, 2000.

Jeffrey L. Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, 48:781–799, 1993.

Tieliang Gong, Qian Zhao, Deyu Meng, and Zongben Xu. Why curriculum learning & self-paced learning work in big/noisy data: A theoretical perspective. *Big Data & Information Analytics*, 1 (1):111–127, 2016.

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

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of Wasserstein GANs. In *arXiv:1704.00028*, 2017.

Kaiming He, Xiangyu Zhang, Shaojing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 770–778, 2016.

Ari Heljakka, Arno Solin, and Juho Kannala. Pioneer networks: Progressively growing generative autoencoder. In *arXiv:1807.03026*, 2018.

Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

Niels Justesen, Ruben Rodriguez Torrado, Philip Bontrager, Ahmed Khalifa, Julian Togelius, and Sebastian Risi. Illuminating generalization in deep reinforcement learning through procedural level generation. *arXiv:1806.10729*, 2018.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018a.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv:1812.04948*, 2018b.

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *arXiv:1807.03039*, 2018.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the 2th International Conference on Learning Representations (ICLR)*, 2013.

Dimitrios Korkinof, Tobias Rijken, Michael O'Neill, Joseph Yearsley, Hugh Harvey, and Ben Glocker. High-resolution mammogram synthesis using progressive generative adversarial networks. In *arXiv:1807.03401*, 2018.

Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Unsupervised learning of hierarchical representations with convolutional deep belief networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. *arXiv:1611.04076*, 2016.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? *arXiv:1801.04406*, 2018.

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

Mark Newman. *Networks: An Introduction*. Oxford, 1st edition, 2010.

Pierre-Yves Oudeyer, Frederic Kaplan, and Verena V. Hafner. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11(2):265–286, 2007.

Ofir Press, Amir Bar, Ben Bogin, Jonathan Berant, and Lior Wolf. Language generation with recurrent generative adversarial networks without pre-training. *arXiv:1706.01399*, 2017.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, 2016.

Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Pal, and Aaron Courville. Adversarial generation of natural language. *arXiv:1705.10929*, 2017.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning (ICML)*, pp. 1530–1538, 2015.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning (ICML)*, pp. 1278–1286, 2014.

Steven Roman. *Introduction to Coding and Information Theory*. Springer, 1st edition, 1996.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3): 211–252, 2015.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. *arXiv:1606.03498*, 2016.

Dietrich Stauffer and Ammon Aharony. *Introduction To Percolation Theory*. CRC Press, 2st edition, 1994.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A generative model for raw audio. In *arXiv:1609.03499*, 2016a.

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *arXiv:1601.06759*, 2016b.

Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C. Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis. Parallel WaveNet: Fast high-fidelity speech synthesis. In *arXiv:1711.10433*, 2017.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 1st edition, 2018.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv:1506.03365*, 2015.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, 2014.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929, 2016.

Will Zou and James McClelland. Progressive development of the number sense in a deep neural network. In *Annual Conference of the Cognitive Science Society*, 2013.

A Appendix

A.1 Centrality measure

The centrality or clustering coefficient pertaining to a cluster in data points or a community in a complex network is a well-studied traditional topic in machine learning and complex systems. Here we introduce the graph-theoretic centrality for the utilization of cluster curriculum. Firstly, we construct a directed graph (digraph) with K nearest neighbors. The weighted adjacency matrix $\boldsymbol{W}$ of the digraph can be formed in this way: $W_{ij} = \exp(-d_{ij}^2/\sigma^2)$ if $\boldsymbol{x}_j$ is one of the nearest neighbors of $\boldsymbol{x}_i$ and 0 otherwise, where d_{ij} is the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ and σ is a free parameter.

The density of data points can be quantified with stationary probability distribution of a Markov chain. For a digraph built from data, the transition probability matrix can be derived by row normalization, say, $P_{ij} = W_{ij}/\sum_j W_{ij}$. Then the stationary probability $\boldsymbol{u}$ can be obtained by solving an eigenvalue problem

$$\boldsymbol{P}^\top \boldsymbol{u} = \boldsymbol{u}, \tag{5}$$

where $\top$ denotes the matrix transpose. It is straightforward to know that $\boldsymbol{u}$ is the eigen-vector of $\boldsymbol{P}^\top$ corresponding to the largest eigenvalue (i.e. 1). $\boldsymbol{u}$ is also defined as a kind of PageRank in many scenarios.

For density-based cluster curriculum, the centrality $\boldsymbol{c}$ coincides with the stationary probability $\boldsymbol{u}$. Figure 1 in the main context shows the plausibility of using the stationary probability distribution to quantify data density.

A.2 Theorem 1 and Proof

Theorem 1. *For a set $\mathcal{C} = \{\boldsymbol{x}_i | \boldsymbol{x}_i \in \mathbb{R}^d\}$ of n data points drawn from normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, the ellipsoid E_α of confidence $1-\alpha$ is defined as $\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x} \leq \chi^2_\alpha$, where $\boldsymbol{\Sigma}$ has no zero eigenvalues*

and $\alpha \in [0, 1]$. Let $N(E_\alpha)$ be the number of spheres of radius ε packed in the ellipsoid E_α. Then we can establish

$$N(E_\alpha) = \left(\frac{\chi_\alpha}{\varepsilon}\right)^d \sqrt{\det(\boldsymbol{\Sigma})}. \tag{6}$$

Proof. As explained in the main context, the ellipse equation with respect to the confidence α can be expressed by the following equation

$$\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x} = \chi_\alpha^2. \tag{7}$$

Suppose that $\lambda_1, \ldots, \lambda_d$ are the eigenvalues of $\boldsymbol{\Sigma}$. Then equation (7) can be written as

$$\frac{x_1^2}{\lambda_1} + \cdots + \frac{x_d^2}{\lambda_d} = \chi_\alpha^2. \tag{8}$$

Further, eliminating χ_α^2 on the right side gives

$$\frac{x_1^2}{\chi_\alpha^2 \lambda_1} + \cdots + \frac{x_d^2}{\chi_\alpha^2 \lambda_d} = 1. \tag{9}$$

Then we derive the length of semi-axis with respect to χ_α, i.e.

$$r_i = \chi_\alpha \sqrt{\lambda_i}. \tag{10}$$

For a d-dimensional ellipsoid E, the volume of E is

$$\text{Vol}(E_\alpha) = \frac{2}{d} \frac{\pi^{d/2}}{\Gamma(d/2)} r_1 r_2 \cdots r_d, \tag{11}$$

where r_i the leng of semi-axis of E and $\Gamma(\cdot)$ is the Gamma function. Substituting (10) into the above equation, we obtain the final formula of volume

$$\text{Vol}(E_\alpha) = \frac{2}{d} \frac{\pi^{d/2}}{\Gamma(d/2)} \chi_\alpha^d \prod_{i=1}^{d} \sqrt{\lambda_i}, \tag{12}$$

$$= \frac{2}{d} \frac{\pi^{d/2}}{\Gamma(d/2)} \chi_\alpha^d \sqrt{\det(\boldsymbol{\Sigma})}. \tag{13}$$

Using the volume formula in (11), it is straightforward to get the volume of packing sphere S_ε

$$\text{Vol}(S_z) = \frac{2}{d} \frac{\pi^{d/2}}{\Gamma(d/2)} \sqrt{\det(\varepsilon^2 \boldsymbol{I})}. \tag{14}$$

By the definition of $N(E_\alpha)$, we can write

$$N(E_\alpha) = \# \text{ of spheres} = \text{Vol}(E_\alpha)/\text{Vol}(S_\varepsilon) \tag{15}$$

$$= \left(\frac{\chi_\alpha}{\varepsilon}\right)^d \sqrt{\det(\boldsymbol{\Sigma})}. \tag{16}$$

We conclude the proof of the theorem. □

A.3 PROCEDURES OF ALGORITHM 1 AND ALGORITHM 2

Algorithm 1 Cluster Curriculum for Generative Models

Require:
$\mathcal{X}$, dataset containing m data points
`GenerativeModel()`, generative models
`QualityScore()`, metric for generative results
l, number of subsets

Ensure:
// Solve centralities
$\boldsymbol{c} = \texttt{Centrality}(\mathcal{X})$ ▷ section A.1
// Cluster curriculum
Get sorted data $\overrightarrow{\mathcal{X}}$ according to descending order of $\boldsymbol{c}$
Divide $\overrightarrow{\mathcal{X}}$ to be $\{\overrightarrow{\mathcal{X}}_0, \overrightarrow{\mathcal{X}}_1, \ldots, \overrightarrow{\mathcal{X}}_l\}$
// Train generative models
for $\overrightarrow{\mathcal{X}}_i \in \{\overrightarrow{\mathcal{X}}_0, \overrightarrow{\mathcal{X}}_1, \ldots, \overrightarrow{\mathcal{X}}_l\}$ **do**
    Initialize model parameters $\boldsymbol{\theta}$ randomly
    $\boldsymbol{\theta}_i = \texttt{GenerativeModel}(\overrightarrow{\mathcal{X}}_0, \boldsymbol{\theta})$ ▷ e.g. GAN
    $\overrightarrow{\mathcal{X}}_0 \leftarrow \{\overrightarrow{\mathcal{X}}_0 \cup \overrightarrow{\mathcal{X}}_i\}$
end for
// Search the optimal set
for $\boldsymbol{\theta}_i \in \{\theta_1 \ldots, \theta_{l+1}\}$ **do**
    Generate data $\overline{\mathcal{X}}$ with model of parameter $\boldsymbol{\theta}_i$
    $s_i = \texttt{QualityScore}(\ \overline{\mathcal{X}}\ )$ ▷ e.g. FID
end for
$i^* = \arg\min s_i, i = 1, \ldots, l+1$
Return the optimal model parameter $\boldsymbol{\theta}_{i^*}$

Algorithm 2 Cluster Curriculum with Active Set

Require:
$\mathcal{X}$, `GenerativeModel()`, `QualityScore()` as in Algorithm 1
$|\mathcal{A}|$, cardinality of active set $\mathcal{A}$
Ensure:
// Solve centralities
$\boldsymbol{c} = \texttt{Centrality}(\mathcal{X})$ ▷ section A.1
// Cluster curriculum
Get sorted data $\overrightarrow{\mathcal{X}}$ according to descending order of $\boldsymbol{c}$
Divide $\overrightarrow{\mathcal{X}}$ to be $\{\overrightarrow{\mathcal{X}}_0, \overrightarrow{\mathcal{X}}_1, \ldots, \overrightarrow{\mathcal{X}}_l\}$
// Train generative models
Initialize model parameters $\boldsymbol{\theta}$ randomly
$\boldsymbol{\theta}_0 = \texttt{GenerativeModel}(\overrightarrow{\mathcal{X}}_0, \boldsymbol{\theta})$ ▷ e.g. GAN
for $\overrightarrow{\mathcal{X}}_i \in \{\overrightarrow{\mathcal{X}}_1, \ldots, \overrightarrow{\mathcal{X}}_l\}$ **do**
    Derive $\overrightarrow{\mathcal{A}}_0$ by randomly sampling $\overrightarrow{\mathcal{X}}_0$
    $\overrightarrow{\mathcal{A}}_0 \leftarrow \{\overrightarrow{\mathcal{A}}_0 \cup \overrightarrow{\mathcal{X}}_i\}$
    // Use pre-training model
    $\boldsymbol{\theta}_i = \texttt{GenerativeModel}(\overrightarrow{\mathcal{A}}_0, \boldsymbol{\theta}_{i-1})$
    $\overrightarrow{\mathcal{X}}_0 \leftarrow \{\overrightarrow{\mathcal{X}}_0 \cup \overrightarrow{\mathcal{X}}_i\}$
end for
// Search the optimal set
for $\boldsymbol{\theta}_i \in \{\theta_1 \ldots, \theta_{l+1}\}$ **do**
    Generate data $\overline{\mathcal{X}}$ with model of parameter $\boldsymbol{\theta}_i$ by sampling a prior ▷ e.g. Gaussian
    $s_i = \texttt{QualityScore}(\overline{\mathcal{X}})$ ▷ e.g. FID
end for
$i^* = \arg\min s_i, i = 1, \ldots, l+1$
Return the optimal model parameter $\boldsymbol{\theta}_{i^*}$