Adaptive IMLE for Few-shot Pretraining-free Generative Modelling

Mehran Aghabozorgi [1] **Shichong Peng** [1] **Ke Li** [1]

Abstract

Despite their success on large datasets, GANs have been difficult to apply in the few-shot setting, where only a limited number of training examples are provided. Due to mode collapse, GANs tend to ignore some training examples, causing overfitting to a subset of the training dataset, which is small in the first place. A recent method called Implicit Maximum Likelihood Estimation (IMLE) is an alternative to GAN that tries to address this issue. It uses the same kind of generators as GANs but trains it with a different objective that encourages mode coverage. However, the theoretical guarantees of IMLE hold under a restrictive condition that the optimal likelihood at all data points is the same. In this paper, we present a more generalized formulation of IMLE which includes the original formulation as a special case, and we prove that the theoretical guarantees hold under weaker conditions. Using this generalized formulation, we further derive a new algorithm, which we dub Adaptive IMLE, which can adapt to the varying difficulty of different training examples. We demonstrate on multiple few-shot image synthesis datasets that our method significantly outperforms existing methods. Our code is available at https://github.com/mehranagh20/AdaIMLE.

1. Introduction

Image synthesis has achieved significant progress over the past decade with the emergence of deep learning. Deep generative models such as GANs (Goodfellow et al., 2014; Brock et al., 2019; Karras et al., 2019; 2020; 2021), VAEs (Kingma & Welling, 2013; Vahdat & Kautz, 2020; Child, 2021; Razavi et al., 2019), diffusion models (Dhariwal & Nichol, 2021; Ho et al., 2020; Karras et al., 2022), score-based models (Song et al., 2021; Song & Ermon, 2019), normalizing flows (Dinh et al., 2017; Kobyzev et al., 2021; Kingma & Dhariwal, 2018), and autoregressive models (Salimans et al., 2017; van den Oord et al., 2016b;a) have made incredible improvements in generated image quality, which makes it possible to generate photorealistic images using these models.

Many of these deep generative models require training on large-scale datasets to produce high-quality images. However, there are many real-life scenarios in that only a limited number of training examples are available, such as orphan diseases in the medical domain and rare events for training autonomous driving agents. One way to address this issue is by fine-tuning a model pre-trained on large auxiliary datasets from similar domains (Wang et al., 2020; Zhao et al., 2020a; Mo et al., 2020). Nonetheless, a large auxiliary dataset with a sufficient degree of similarity to the task at hand may not be available in all domains. If an insufficient similar auxiliary dataset were used regardless, image quality may be adversely impacted, as shown in (Zhao et al., 2020b). Therefore, there have been efforts in tackling the challenging setting of few-shot unconditional image synthesis without auxiliary pre-training (Liu et al., 2021; Kong et al., 2022; Li et al., 2022), and we will focus on this setting.

The scarcity of training data in this setting makes it especially important for generative models to make full use of all training examples. This requirement sets it apart from the many-shot setting with abundant training data, where ignoring some training examples does not cause as big an issue. As a result, despite achieving impressive performance in the many-shot setting, GANs are challenging to apply to the few-shot setting due to the well-known problem of mode collapse, where the generator only learns from a subset of the training images and ignores the rest. A recent work (Li & Malik, 2018) proposed an alternative technique called Implicit Maximum Likelihood Estimation (IMLE) for unconditional image synthesis. Similar to GAN, IMLE uses a generator, but rather than adopting an adversarial objective which encourages each generated image to be similar to some training images, IMLE encourages each training image to have some similar generated images. Therefore, the generated images could cover all training examples without

[1]APEX Lab, School of Computing Science, Simon Fraser University. Correspondence to: Mehran Aghabozorgi <mehran_aghabozorgi@sfu.ca>, Shichong Peng <shichong_peng@sfu.ca>, Ke Li <keli@sfu.ca>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

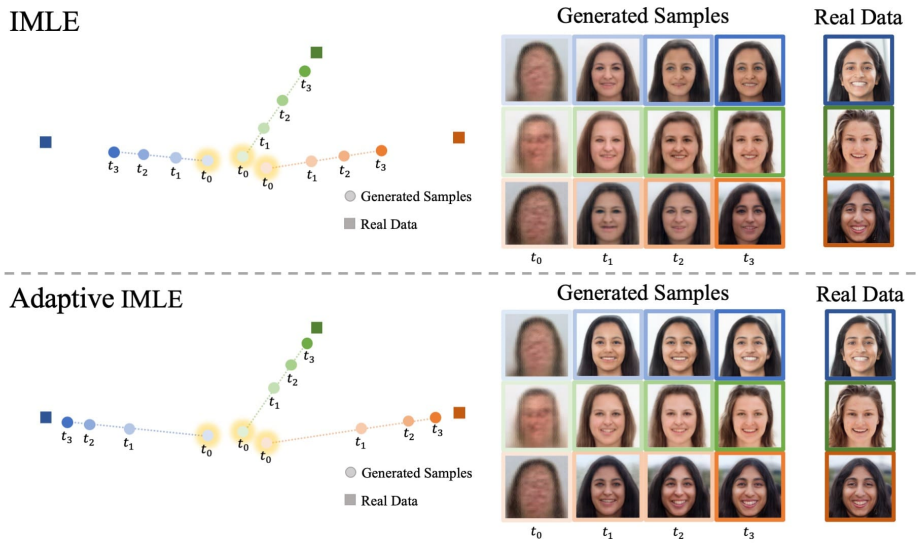

Figure 1: Schematic illustration that compares vanilla IMLE (Li & Malik, 2018) (top row) with the proposed algorithm, Adaptive IMLE (bottom row). While IMLE treats all training examples (denoted by the squares on the left) equally and pulls the generated samples (denoted by the circles on the left) towards them at a uniform pace, Adaptive IMLE adapts to the varying difficulty of each training example and pulls the generated samples towards them at an individualized pace that depends on the training example. The dashed line on the left figure illustrates the progression towards three data points at four comparable epochs (denoted as t_0 to t_3) with the starting positions highlighted. The corresponding generated samples are shown on the right. As shown, Adaptive IMLE can converge to the various data points faster and closer than IMLE.

collapsing to a subset of the modes.

However, the theoretical guarantees of IMLE hold under a restrictive condition that all data points should have an identical optimal likelihood. The IMLE algorithm, therefore, treats all training examples equally when optimizing the model parameters and ignores the varying difficulty in learning from different training examples. As shown in the top row of Fig. 1, the generated samples make uneven progress toward different training examples using IMLE, leading to overfitting to some examples and underfitting to others. While this may not cause a major issue in the many-shot setting because many data points are expected to have similar optimal likelihoods, it can be quite problematic in the few-shot setting, since uneven fitting can impact the model quality substantially due to the small total number of training examples that the model is trained on.

In this paper, we introduce a generalized formulation of IMLE, which in turn enables the derivation of a new algorithm that requires fewer conditions and gets around the aforementioned issue. In particular, we mathematically prove that the theoretical guarantees of the generalized formulation hold under weaker conditions and subsumes the IMLE formulation as a special case. Furthermore, we derive an algorithm called *Adaptive IMLE* using this generalized formulation, which could adapt to points with different difficulties, as illustrated in the bottom row of Fig. 1. Furthermore, we conducted experiments on six datasets to evaluate the performance of our method compared to prior few-shot image synthesis baselines. Our results demonstrate that our method achieves significant improvements in terms of both image fidelity and mode coverage, establishing a new state-of-the-art.

2. Related Work

There are two broad families of work on few-shot learning, one that focuses on discriminative tasks such as classification (O'Mahony et al., 2019; Finn et al., 2017; Snell et al., 2017) and another that focuses on generative tasks. In this paper, we focus on the latter. Similar to many-shot generation tasks, few-shot generation tasks take a limited number of training examples as input and aim to generate samples that are similar to those training examples. What is different from the many-shot setting is that it is crucial for the generative model to utilize all the training examples in the few-shot setting. Due to the scarcity of available data for training, ignoring even just a few data points would cause a more serious issue in the few-shot setting than in the many-shot setting. One line of work focuses on pre-training on large-scale auxiliary datasets from similar domains and adapting the pre-trained models for the few-shot task. This has been applied to unconditional image generation (Li et al., 2020; Zhao et al., 2020a; Mo et al., 2020; Ojha et al., 2021; Wang

et al., 2020), conditional image generation (Sinha et al., 2021; Liu et al., 2019) and video generation (Wang et al., 2019). However, there are no guarantees on the existence of such large-scale auxiliary datasets for all domains, and recent studies (Zhao et al., 2020b; Kong et al., 2022) also showed that fine-tuning from a dissimilar domain could even lead to the degradation of generated image quality.

In this paper, we focus on the setting without fine-tuning pre-trained models from auxiliary datasets. Most prior work considered applying GANs to this setting and developed methods for alleviating the well-known mode collapse problem of GANs. FastGAN (Liu et al., 2021) introduced a skip-layer excitation module for faster training and used self-supervision for the discriminator to learn more descriptive features, which aids in better mode coverage of the generator. MixDL (Kong et al., 2022) introduced a two-sided distance regularization to facilitate learning smooth and mode-preserving latent space. FakeCLR (Li et al., 2022) aims to improve image synthesis quality by introducing extensive data augmentation and applying contrastive learning only on perturbed fake samples. Despite these improvements, some degree of mode collapse still remains. A recent method called Implicit Maximum Likelihood Estimation (IMLE) (Li & Malik, 2018) adopted a different objective function and showed promising results towards alleviating mode collapse on unconditional image synthesis tasks. Prior IMLE-based methods mainly focused on conditional image synthesis (Li* et al., 2020; Peng et al., 2022). In this work, we build on (Li & Malik, 2018) and introduce a novel and more generalized formulation of IMLE to make it more suitable for the unconditional few-shot setting.

3. Background: Implicit Maximum Likelihood Estimation (IMLE)

In unconditional image synthesis, the goal is to learn the underlying probability distribution of images, denoted as $p(\mathbf{x})$, which allows us to generate new synthesized images by sampling from this distribution. Implicit generative models are commonly used for this task, such as the generator in GANs. The generator, parameterized as a neural network with parameters θ, maps latent codes $\mathbf{z}$ drawn from a standard Gaussian distribution $\mathcal{N}(0, \mathbf{I})$ to generate images $\mathbf{x}$. One way to learn this model is with the GAN objective, which introduces a discriminator that aims to distinguish between generated images $T_\theta(\mathbf{z})$ and real images $\mathbf{x}$. The generator is trained to produce more realistic images that would fool the discriminator. However, the output $T_\theta(\mathbf{z})$ tends to recover only a subset of the training examples even when varying all values of $\mathbf{z}$. This issue is known as mode collapse, and the intuitive reason behind it is that the adversarial objective of GAN only encourages each generated sample to be similar to some training examples, but there is no guarantee that all training examples will have some similar generated samples. In the few-shot image synthesis setting, the issue of mode collapse is even more significant given the limited number of training examples that are available in the first place.

A more recent method known as Implicit Maximum Likelihood Estimation (IMLE) (Li & Malik, 2018) proposed an alternative objective to address this issue. Instead of making each generated sample similar to *some* training examples, IMLE tries to ensure that samples can be generated around *each* training example $\mathbf{x}_i$. The generator T_θ is encouraged to pull some samples $T_\theta(\mathbf{z}_j)$ towards each $\mathbf{x}_i$, thereby rewarding coverage of the modes associated with all training examples.

More precisely, the IMLE objective takes the following form:

$$\min_\theta \mathbb{E}_{z_1,\ldots,z_m \sim \mathcal{N}(0,I)} \left[\sum_{i=1}^{n} \min_{j \in [m]} d\left(\mathbf{x}_i, T_\theta(\mathbf{z}_j)\right) \right] \quad (1)$$

where $d(.,.)$ is a distance metric, m is a hyperparameter, and $\mathbf{x}_i$ is the i^{th} training example. The training procedure involves finding the latent code $\mathbf{z}_i^*$ that produces the nearest generated sample to each training example $\mathbf{x}_i$, and optimizing the model parameter θ by minimizing the distance from the selected sample $T_\theta\left(\mathbf{z}_i^*\right)$ to the target data $\mathbf{x}_i$. Detailed pseudocode of the algorithm can be found in Algorithm 1.

Algorithm 1 Vanilla IMLE procedure

Require: The set of inputs $\{\mathbf{x}_i\}_{i=1}^n$
Initialize the parameters θ of the generator T_θ
for $k = 1$ **to** K **do**
  Pick a random batch $S \subseteq [n]$
  Draw latent codes $Z \leftarrow \mathbf{z}_1, ..., \mathbf{z}_m$ from $\mathcal{N}(0, \mathbf{I})$
  $\mathbf{z}_i^* \leftarrow \arg\min_{\mathbf{z}_j \in Z} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)) \; \forall i \in S$
  for $l = 1$ **to** L **do**
    Pick a random mini batch $\tilde{S} \subseteq S$
    $\theta \leftarrow \theta - \eta \nabla_\theta \left(\sum_{i \in \tilde{S}} d\left(\mathbf{x}_i, T_\theta\left(\mathbf{z}_i^*\right)\right) \right) / |\widetilde{S}|$
  end for
end for
return θ

Despite the algorithm's simplicity, a restrictive condition needs to be satisfied for the theoretical guarantees of IMLE to hold, that is requiring a uniform optimal likelihood for all data points. As an example, consider a dataset with two clusters with the same number of points where one cluster has a large variance and the other has a small variance. In this case, the training examples from the high-variance cluster are more difficult to learn than the training examples from the low-variance cluster, because of the sparser coverage of the space in the former cluster. If we consider what the ground truth data distribution looks like, it is a bimodal distribution, with the mode corresponding to the low-variance

cluster having a higher likelihood than the other. So requiring uniform optimal likelihood for all data points, as IMLE does, will result in overfitting to the low-variance cluster and underfitting to the high-variance cluster, which is not optimal. We refer readers to the IMLE paper (Li & Malik, 2018) for more details.

4. Method

In this paper, we devise a generalized formulation of IMLE, whose theoretical guarantees hold under more general conditions than vanilla IMLE (Sec 4.1). This formulation subsumes vanilla IMLE as a special case and also gives rise to a new algorithm which we call *Adaptive IMLE*. It turns out that Adaptive IMLE offers theoretical and practical advantages over IMLE, which we will demonstrate (Sec 4.2).

4.1. Generalized Formulation

Since T_θ is an implicit generative model, the likelihood induced by the model p_θ cannot in general be expressed in closed form, and so evaluating it numerically is typically computationally intractable. In order to train the generative model, we would like to maximize the likelihood of the training examples without actually needing to evaluate the likelihood. Below we will consider the generalized objective we propose and show that optimizing the objective is equivalent to maximizing the sum of likelihoods at the training examples, without requiring the evaluation of likelihood.

Consider the following optimization problem:

$$\max_\theta \mathcal{L}_{\{\tau_i\}_i}(\theta) := \max_\theta \mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\frac{1}{n}\sum_{i=1}^{n}\frac{1}{w_i}\left(\tau_i - \frac{1}{m}\sum_{j=1}^{m}\Phi_{\tau_i}(d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)))\right)\middle|\{\tau_i\}_i\right] \quad (2)$$

where T_θ, $d(\cdot,\cdot)$ and m are as defined in Eqn 1. For each training example i, there is a weighting term w_i and a threshold τ_i, which may be stochastic. The function $\Phi_\tau(\cdot)$ transforms distances from each training example to each sample from the model. We will choose w_i, τ_i and $\Phi_\tau(\cdot)$ based on the insight revealed by lemmas below.

We will present the high-level sketches of our key lemmas (omitting some technicalities) and delineate their interpretations and significance. The precise statements of the lemmas and their proofs are left to Appendix A.

We will first present a lemma that relates an expectation of a random variable to the weighted integral of one minus its cumulative density function (CDF) evaluated at different points, which we will refer to as cumulative densities.

Lemma 1. *Let X be a non-negative random variable and Φ be a continuous function on $[0,\infty)$. If Φ' is integrable on all closed intervals in $[0,\infty)$,*

$$\mathbb{E}\left[\Phi(X)\right] = \Phi(0) + \int_0^\infty \Phi'(t)\Pr(X \geq t)dt$$

This lemma is useful because the left-hand side (LHS) is easy to approximate with Monte Carlo estimates of expectations, and the right-hand side (RHS) is a weighted integral of one minus cumulative densities, which are intractable to compute in general. It enables us to control the weighting of different cumulative densities by choosing the function Φ.

Recall that our goal is to maximize the likelihood at each training example without actually computing the likelihood. We can leverage Lemma 1 for this purpose, by choosing the non-negative random variable X appropriately. We choose X to be the distance between a training example and a generated sample $d(\mathbf{x}_i, T_\theta(\mathbf{z}_j))$. With this choice, Lemma 1 gives us a way to relate a weighted integral of the average likelihoods within differently sized neighbourhoods around the training example $\mathbf{x}_i$ (RHS) to the expectation of a function of the distance $d(\mathbf{x}_i, T_\theta(\mathbf{z}_j))$ (LHS).

Moreover, we'd like to restrict the average likelihoods we integrate over to only those within neighbourhoods of certain sizes rather than from 0 to ∞. Specifically, we'd like to integrate from $\delta\tau$ to τ, where $\tau > 0$ is the radius of the largest neighbourhood and $0 \leq \delta < 1$ is a tightening threshold. To this end, we can choose the weighting function $\Phi'_\tau(\cdot)$ to be 1 when $\delta\tau \leq t \leq \tau$, and 0 otherwise. One choice of such $\Phi_\tau(\cdot)$ that satisfies this condition and its associated $\Phi'_\tau(\cdot)$ are:

$$\Phi_\tau(t) = \begin{cases} \delta\tau & t < \delta\tau \\ t & \delta\tau \leq t \leq \tau \\ \tau & t > \tau \end{cases} \quad \Phi'_\tau(t) = \begin{cases} 0 & t < \delta\tau \\ 1 & \delta\tau \leq t \leq \tau \\ 0 & t > \tau \end{cases}$$

Using this choice of $\Phi_\tau(\cdot)$, we obtain the following lemma for a particular training example $\mathbf{x}_i$.

Lemma 2. *Under the choice of $\Phi_\tau(\cdot)$ above and its associated $\Phi'_\tau(\cdot)$,*

$$\begin{aligned}&\mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\Phi_{\tau_i}(d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)))\middle|\{\tau_i\}_i\right] \\ =&\tau_i - \int_{\delta\tau_i}^{\tau_i}\Pr(d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)) < t)dt.\end{aligned}$$

This lemma shows that, for one training example $\mathbf{x}_i$, the expectation on the LHS reduces to τ_i minus the integral of

the average likelihoods within balls whose radii lie between $\delta\tau_i$ and τ_i. Applying Lemma 2 to all training examples $\mathbf{x}_1, \ldots, \mathbf{x}_n$, we obtain the following lemma that reveals what the overall objective in Eqn. 2 optimizes.

Lemma 3. *Under the choice of $\Phi_\tau(\cdot)$ above and its associated $\Phi'_\tau(\cdot)$,*

$$\mathcal{L}_{\{\tau_i\}_i}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{mw_i}\sum_{j=1}^{m}\int_{\delta\tau_i}^{\tau_i}\Pr(d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)) < t)dt.$$

Lemma 3 shows that $\mathcal{L}_{\{\tau_i\}_i}(\theta)$ implicitly computes the average likelihood that the generative model assigns to the neighbourhood of each data point, and τ_i controls the radius of the neighbourhood size. Since we would like to maximize probability in the immediate neighbourhood of each data point, we would like τ_i to be small. So should we choose an arbitrarily small value for τ_i? Recall that by definition of $\Phi_{\tau_i}(\cdot)$, if $d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)) > \tau_i$, $\Phi_{\tau_i}(d(\mathbf{x}_i, T_\theta(\mathbf{z}_j))) = \tau_i$. So, for a very small τ_i, it may well be the case that $d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)) > \tau_i \ \forall j$, which would make the Monte Carlo estimate of $\mathcal{L}_{\{\tau_i\}_i}(\theta)$, i.e., $\frac{1}{n}\sum_{i=1}^{n}\frac{1}{w_i}\left(\tau_i - \frac{1}{m}\sum_{j=1}^{m}\Phi_{\tau_i}(d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)))\right)$, zero. Since this is a constant, the gradient w.r.t. the parameters is zero, which makes gradient-based learning impossible. This would happen whenever $\tau_i < \min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j))$, and so the smallest τ_i that can be chosen is $\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j))$ (which is treated as a constant rather than a function of θ).

With this choice of τ_i, assuming that there is a unique j^* such that $d(\mathbf{x}_i, T_\theta(\mathbf{z}_{j^*})) = \min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j))$ (which happens almost surely), the objective can be simplified to:

$$\mathcal{L}_{\{\tau_i\}_i}(\theta) := \mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\frac{1}{nm}\sum_{i=1}^{n}\frac{1}{w_i}\left(\tau_i - \max(\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)), \delta\tau_i)\right)\middle|\{\tau_i\}_i\right] \quad (3)$$

If we minimize the objective in Eqn. 3, we get a novel objective known as the Adaptive IMLE objective. The solution to the Adaptive IMLE objective can be expressed as:

$$\begin{aligned}&\arg\max_\theta \mathcal{L}_{\{\tau_i\}_i}(\theta)\\ &= \arg\min_\theta \mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\sum_{i=1}^{n}\frac{1}{w_i}\max(\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)), \delta\tau_i)\middle|\{\tau_i\}_i\right] \quad (4)\end{aligned}$$

It turns out that the vanilla IMLE objective can be recovered as a special case, by choosing $\delta = 0$ and $w_1 = w_2 = \cdots = w_n$.

$$\begin{aligned}&\arg\max_\theta \mathcal{L}_{\{\tau_i\}_i}(\theta)\\ &= \arg\min_\theta \mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\sum_{i=1}^{n}\frac{1}{w_i}\max(\min_{j\in[m]} d\left(\mathbf{x}_i, T_\theta(\mathbf{z}_j)\right), 0)|\{\tau_i\}_i\right]\\ &= \arg\min_\theta \mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\sum_{i=1}^{n}\frac{1}{w_i}\min_{j\in[m]} d\left(\mathbf{x}_i, T_\theta(\mathbf{z}_j)\right)\right]\\ &= \arg\min_\theta \mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\sum_{i=1}^{n}\min_{j\in[m]} d\left(\mathbf{x}_i, T_\theta(\mathbf{z}_j)\right)\right]\end{aligned}$$

4.1.1. Curriculum Learning

Recall that our goal is to maximize the likelihood of the immediate neighbourhood around each data point, and the size of this neighbourhood is controlled by τ_i. Therefore, we want to make τ_i small. In order to make τ_i small without impeding learning, we need to make $\mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}[\tau_i] = \mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j))\right]$ small. To this end, we can either increase m, the number of samples, or train T_θ so that the samples it produces are close to the data point $\mathbf{x}_i$. The former is computationally expensive, and so we will devise a method to achieve the latter.

We propose a curriculum learning strategy, which solves a sequence of optimization problems with different τ_i's, such that τ_i's get smaller for optimization problems later in the sequence. The earlier optimization problems in the sequence help train T_θ to produce samples close to the data points. After each optimization problem is solved to convergence, we start solving the next optimization problem with θ initialized to the solution found previously.

This will make τ_i's smaller and smaller. If they eventually converge to zero, then it turns out that we would have equivalently maximized the sum of likelihoods $p_\theta(\mathbf{x}_i)$ of the training examples under the probability distribution induced the generative model, as shown in the lemma below.

Lemma 4. *Suppose p_θ is continuous at all data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, under the choice of $w_i = \int_{\delta\tau_i}^{\tau_i}\mathrm{vol}(B_t(\mathbf{x}_i))dt := \int_{\delta\tau_i}^{\tau_i}\int_{B_t(\mathbf{x}_i)}d\mathbf{x}dt$, where $B_r(\mathbf{x}) = \{\mathbf{y}|d(\mathbf{y},\mathbf{x}) < r\}$ is an open ball of radius r centred at $\mathbf{x}$,*

$$\lim_{\{\tau_i\to 0^+\}_i}\mathcal{L}_{\{\tau_i\}_i}(\theta) = \frac{1}{n}\sum_{i=1}^{n}p_\theta(\mathbf{x}_i)$$

Here, w_i is a normalizing factor for neighbouring regions around each data point $\mathbf{x}_i$. For common metrics, such as ℓ_p distances, w_i can be found in closed form, details are included in Appendix A. This lemma shows the theoretical guarantees of Adaptive IMLE hold under more general conditions that those required by vanilla IMLE.

4.2. Adaptive IMLE

The key difference from the objective in Eqn. 4 to the original IMLE formulation in Eqn. 1 is the individualized neighbourhood radius τ_i around each data point $\mathbf{x}_i$. This change in the objective is crucial, as it allows the model to adapt to the varying difficulty in learning different training examples, hence the algorithm name, *Adaptive IMLE*.

As mentioned in Sect. 4.1.1, we need to gradually decrease τ_i in order to make the learning feasible. To achieve this, we adopt a progressive approach by training T_θ in stages, each aimed at making incremental improvements to T_θ. As we continue to train the model, the generated samples produced by T_θ progressively get closer to the data points, thereby decreasing $\tau_i = \min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j))$.

During each stage of training, the algorithm optimizes the model parameters until the distance between the generated sample and the target data $d(\mathbf{x}_i, T_\theta(\mathbf{z}_j))$ decreases to $\delta\tau_i$. Here δ is a tightening coefficient that determines the required progress of the selected sample towards each training example at each optimization stage.

Upon reaching the threshold, the algorithm generates new samples and updates the threshold τ_i to be the distance between the data point and its nearest sample among the *newly* generated samples, i.e., $\tau_i = \min_{j'\in[m]_{\text{new}}} d(\mathbf{x}_i, T_\theta(\mathbf{z}_{j'}))$. This updated threshold $\delta\tau_i$ then serves as the new target for learning $\mathbf{x}_i$.

Now let's turn our attention to the optimization problem in each stage of the curriculum. Specifically, consider the weighted objective in Eqn. 4. We denote the distance threshold at the k^{th} iteration as $\{\tau_i^k\}_i$ and we assume uniform convergence across all the $\{\tau_i^k\}_i$ $\forall i \in [n]$, i.e., $\text{plim}_{k\to\infty} \max_{i\in[n]}(\tau_i^k) = 0$, which we usually observe in practice.

At the beginning of each optimization problem, τ_i is set to the value of $\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j))$. Since $0 \le \delta < 1$ and $\tau_i^k \ge 0$, $\max(\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)), \delta\tau_i^k)$ is lower-bounded by $\delta\tau_i^k$ and upper-bounded by τ_i^k. Furthermore, we have $\text{plim}_{k\to\infty}\delta\tau_i^k = \text{plim}_{k\to\infty}\tau_i^k = 0$. By squeeze theorem $\text{plim}_{k\to\infty} \max(\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)), \delta\tau_i^k) = 0$.

Note that the weighting term $\frac{1}{w_i}$ in Eqn. 4 is fixed for all data points x_i, this means that the objective value of the weighted objective would converge to $\text{plim}_{k\to\infty} \sum_{i=1}^n \frac{1}{w_i} \max(\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)), \delta\tau_i^k) = 0$.

Now let's consider an unweighted optimization problem $\tilde{\mathcal{L}}_{\{\tau_i^{k'}\}_i}$, which is a variant to the objective in Eqn. 4 without the $\frac{1}{w_i}$ factor:

Algorithm 2 Adaptive IMLE Procedure

Require: The set of inputs $\{\mathbf{x}_i\}_{i=1}^n$, tightening coefficient $\delta \in [0, 1)$
Initialize the parameters θ of the generator T_θ
Draw latent codes $Z \leftarrow \mathbf{z}_1, ..., \mathbf{z}_m$ from $\mathcal{N}(0, \mathbf{I})$
$\mathbf{z}_i^* \leftarrow \arg\min_{\mathbf{z}_j \in Z} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j))$
$\tau_i \leftarrow d\left(\mathbf{x}_i, T_\theta\left(\mathbf{z}_i^*\right)\right)\ \forall i \in [n]$
for $k = 1$ **to** K **do**
  Pick a random batch $S \subseteq [n]$
  $\theta \leftarrow \theta - \eta\nabla_\theta \left(\sum_{i\in S} d\left(\mathbf{x}_i, T_\theta\left(\mathbf{z}_i^*\right)\right)\right)/|S|$
  Draw latent codes $Z \leftarrow \mathbf{z}_1, ..., \mathbf{z}_m$ from $\mathcal{N}(0, \mathbf{I})$
  for $i \in S$ **do**
    if $d\left(\mathbf{x}_i, T_\theta\left(\mathbf{z}_i^*\right)\right) \le \delta\tau_i$ **then**
      $\mathbf{z}_i^* \leftarrow \arg\min_{\mathbf{z}_j \in Z} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j))$
      $\tau_i \leftarrow d(\mathbf{x}_i, T_\theta(\mathbf{z}_i^*))$ {Update the threshold}
    end if
  end for
end for
return θ

$$\arg\min_\theta \mathbb{E}_{z_1,...,z_m\sim\mathcal{N}(0,I)} \left[\sum_{i=1}^n \max(\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)), \delta\tau_i^{k'}) \,\middle|\, \{\tau_i^{k'}\}_i\right] \quad (5)$$

Similar to the weighted case, the objective value for the unweighted problems $\sum_{i=1}^n \max(\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)), \delta\tau_i^{k'})$ would converge to 0 as $\text{plim}_{k'\to\infty} \max(\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)), \delta\tau_i^{k'}) = 0$.

This means that for any given $\epsilon' > 0$, there exists an optimal solution $\theta^{k'}$ to the unweighted problem such that $\max(\min_{j\in[m]} d(\mathbf{x}_i, T_{\theta^{k'}}(\mathbf{z}_j)), \delta\tau_i^{k'}) < \epsilon'$. Suppose there is a weighted problem such that $\frac{1}{w_i}\max(\min_{j\in[m]} d(\mathbf{x}_i, T_{\theta^k}(\mathbf{z}_j)), \delta\tau_i^k) > \epsilon > 0$, then the solution $\theta^{k'}$ to the unweighted problem can also solve the current weighted problem if we set $\epsilon' = w_i\epsilon > 0$. This is because w_i is positive and $\max(\min_{j\in[m]} d(\mathbf{x}_i, T_{\theta^{k'}}(\mathbf{z}_j)), \delta\tau_i^{k'}) < \epsilon' = w_i\epsilon$, and so $\frac{1}{w_i}\max(\min_{j\in[m]} d(\mathbf{x}_i, T_{\theta^{k'}}(\mathbf{z}_j)), \delta\tau_i^{k'}) < \epsilon$ which is the weighted objective in Eqn. 4. This demonstrates that for any weighted optimization problem, we could always solve it by the optimal solution to an unweighted problem.

Let's examine the other direction. Suppose there is an unweighted problem such that $\max(\min_{j\in[m]} d(\mathbf{x}_i, T_{\theta^{k'}}(\mathbf{z}_j)), \delta\tau_i^{k'}) > \epsilon' > 0$. Similar to the derivation above, we could find a solution θ^k to the weighted problem such that $\frac{1}{w_i}\max(\min_{j\in[m]} d(\mathbf{x}_i, T_{\theta^k}(\mathbf{z}_j)), \delta\tau_i^k) < \epsilon$, where $\epsilon = \frac{1}{w_i}\epsilon'$. Since $\frac{1}{w_i}$ is positive, we have $\max(\min_{j\in[m]} d(\mathbf{x}_i, T_{\theta^k}(\mathbf{z}_j)), \delta\tau_i^k) < w_i\epsilon = \epsilon'$. This shows that θ^k is also an optimal solution to the unweighted problem.

Hence, the optimal solutions for both the weighted and unweighted problems are equivalent. This equivalence holds across all stages of the optimization problems. As a result, the sets of optimizers for the weighted and unweighted objectives are identical. Consequently, optimizing the unweighted objective is equivalent to optimizing the weighted objective. Therefore, we utilize the unweighted objective in our algorithm.

Now, if we putting everything together, we obtain the Adaptive IMLE algorithm shown in Algorithm 2. In practice, we make slight modifications to improve efficiency, and the details are provided in Appendix B. We further examine the convergence of τ_i's by plotting the changes in their values during the training process, as shown in Fig. 2. The results demonstrate that the maximum value of τ_i converges to zero, validating our assumption of uniform convergence. Furthermore, as the training progresses, we observe a trend of increased concentration in the values of τ_i's, indicating a more balanced convergence among different data points achieved by our algorithm.

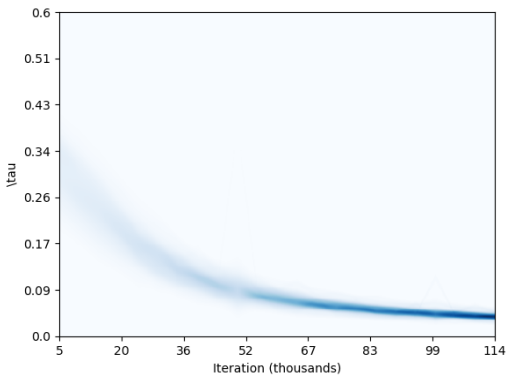

Figure 2: Evolution of τ_i values during training. Each slice of x axis shows the concentration of τ_i values in that specific iteration. Darker colour indicates higher concentration. The results illustrate the uniform convergence of τ_i towards zero, validating our assumption. Additionally, the values of τ_i's become more concentrated as the model trains, demonstrating a more even convergence towards different data points.

5. Experiments

Baselines We compare our method to recent few-shot unconditional image synthesis methods that operate in the same setting we consider, namely without needing to pretrain on auxiliary datasets. Three of such recent methods are FastGAN (Liu et al., 2021), MixDL (Kong et al., 2022) and FakeCLR (Li et al., 2022). We also compare our method to the recent diffusion model EDM (Karras et al., 2022) to demonstrate its potential impact in a broader context.

Table 1: Comparison of image quality, as measured by FID (Heusel et al., 2017), between real data and 5000 randomly generated samples from each method. Lower FID value is better. Our method outperforms all baselines.

	Grumpy Cat	Obama	Panda	Cat	Dog	FFHQ subset
	FID↓	FID↓	FID↓	FID↓	FID↓	FID↓
FakeCLR (Li et al., 2022)	<u>20.6</u>	<u>29.9</u>	<u>8.8</u>	<u>27.4</u>	<u>44.4</u>	62.11
FastGAN (Liu et al., 2021)	26.6	41.1	10.0	35.1	50.7	<u>54.2</u>
MixDL (Kong et al., 2022)	24.5	43.4	10.6	56.1	81.2	62.3
EDM (Karras et al., 2022)	36.9	51.3	23.7	48.6	100.1	79.1
Adaptive IMLE (Ours)	**19.1**	**25.0**	**7.6**	**24.9**	**43.0**	**33.2**

Training Details Our network architecture is modified from (Child, 2021), where we keep the decoder architecture and replace the encoder with a fully-connected mapping network inspired by (Karras et al., 2019). We choose an input latent dimension of 1024, $m = 10000$ and a tightening coefficient $\delta = 0.98$. We train our model for less than 200k iterations with a mini-batch size of 4 using the Adam optimizer (Kingma & Ba, 2015) with a learning rate of 2×10^{-6} on a single NVIDIA V100 GPU.

Datasets We evaluate our method and the baselines on a wide range of natural image datasets at 256×256 resolution, which includes Animal-Face Dog and Cat (Si & Zhu, 2012), Obama, Panda, and Grumpy-cat (Zhao et al., 2020b) and Flickr-FaceHQ (FFHQ) subset (Karras et al., 2019). All datasets contain 100 images except for Dog and Cat which contain 389 and 160 images respectively. The FFHQ subset consists of 100 FFHQ images with similar backgrounds, in order to highlight diversity in the generation of foregrounds.

Evaluation Metrics We use the Fréchet Inception Distance (FID) (Heusel et al., 2017) to measure the perceptual quality of the generated images, where we randomly generate 5000 images and compute FID between the generated samples and real images in each dataset. To evaluate the mode modelling accuracy (precision) and coverage (recall), we use the precision metric of (Kynkäänniemi et al., 2019) to measure the former, and use the recall metric of (Kynkäänniemi et al., 2019) and LPIPS backtracking score (Liu et al., 2021) to measure the latter. For LPIPS backtracking, we use 90% of the full dataset for training and evaluate the metric using the remaining 10% of the dataset.

5.1. Quantitative Results

We compare the FID across all methods in Tab. 1. As shown, our method outperforms the baselines in terms of FID on all datasets. We compare the mode accuracy and coverage in Tab. 2. As shown, our method significantly outperforms the baselines in terms of recall while achieving the best or second best precision across all datasets. We also compare the LPIPS backtracking score in Tab. 3. Our

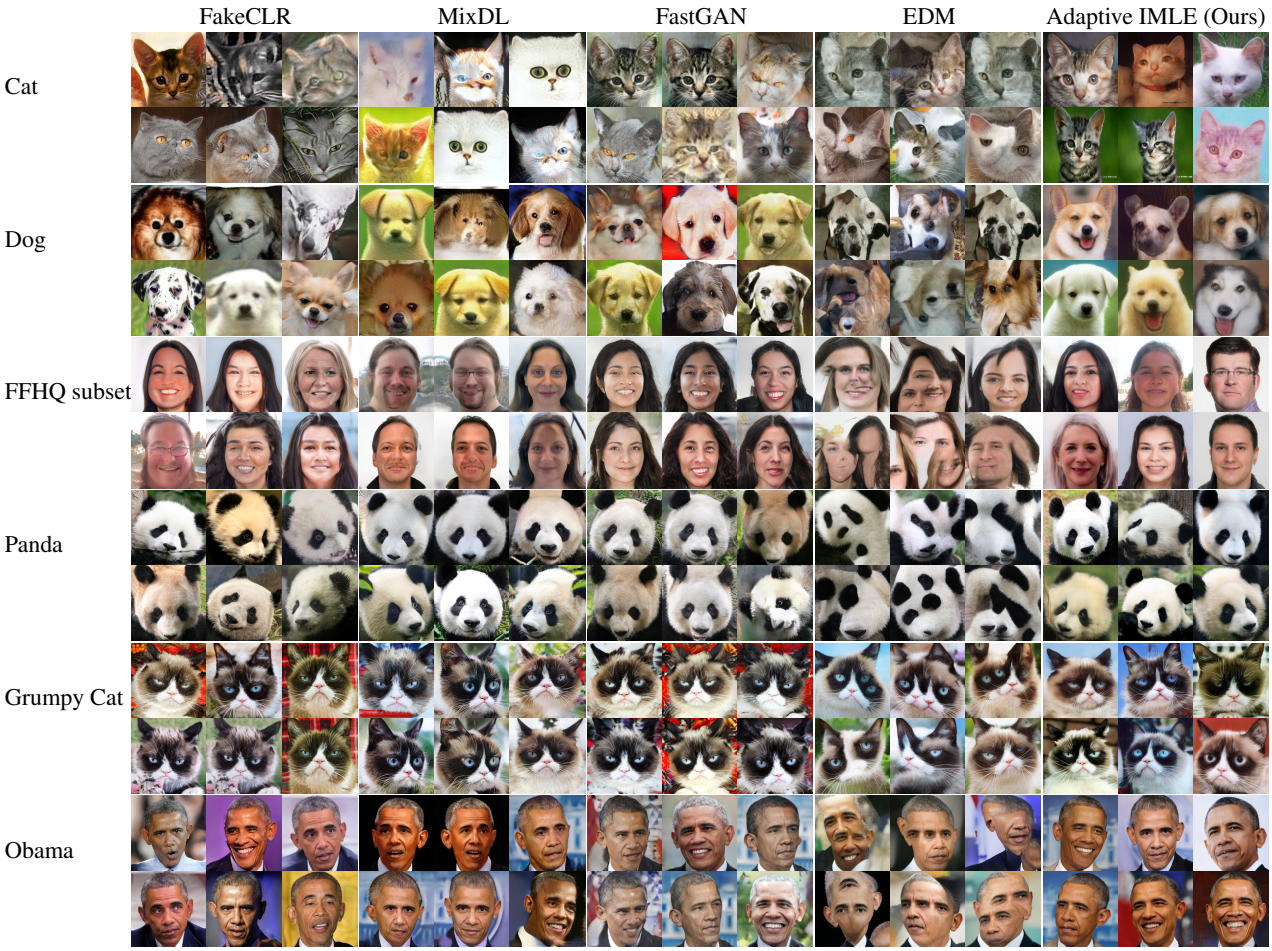

Figure 3: Qualitative comparison of images generated by our method and the baselines, FakeCLR (Li et al., 2022), FastGAN (Liu et al., 2021), MixDL (Kong et al., 2022) and EDM (Karras et al., 2022). Our method shows higher sample quality and diversity. In contrast, the samples generated by the baselines exhibit distortions and limited diversity, as supported by the results in Table 2. These results indicate that the baselines suffer from spurious modes or mode collapse.

Table 2: Precision and recall (Kynkäänniemi et al., 2019) are computed for 1000 randomly generated samples and the target dataset. Higher precision shows better fitting to the target dataset, while higher recall corresponds to better mode coverage. Our method achieves significantly better recall compared to the baselines and scores the best or second best precision across all cases.

	Grumpy Cat		Obama		Panda		Cat		Dog		FFHQ subset	
	Prec.↑	Rec.↑	Prec.↑	Rec.↑	Prec.↑	Rec.↑	Prec.↑	Rec.↑	Prec.↑	Rec.↑	Prec.↑	Rec.↑
FakeCLR (Li et al., 2022)	**0.97**	0.39	0.96	0.30	0.97	0.41	**0.99**	0.55	0.95	0.34	0.71	0.25
FastGAN (Liu et al., 2021)	0.91	0.13	0.92	0.09	0.96	0.16	0.97	0.08	0.96	0.19	0.91	0.13
MixDL (Kong et al., 2022)	0.93	0.35	0.91	0.47	0.93	0.30	0.91	0.50	0.86	0.15	0.77	0.30
Adaptive IMLE (Ours)	**0.97**	**0.72**	**0.99**	**0.68**	**0.98**	**0.63**	0.98	**0.86**	**0.97**	**0.61**	**0.99**	**0.77**

Table 3: LPIPS below represents LPIPS backtracking score (Liu et al., 2021). For this metric, each model is trained on 90% of the dataset. The resulting model is used to backtrack in the latent space and reconstruct the remaining 10%. Lower LPIPS backtracking score shows better mode coverage of the training data. Our method significantly outperfoms the baselines.

	Grumpy Cat	Obama	Panda	Cat	Dog	FFHQ subset
	LPIPS↓	LPIPS↓	LPIPS↓	LPIPS↓	LPIPS↓	LPIPS↓
FakeCLR (Li et al., 2022)	0.467	0.446	0.479	0.570	0.602	0.316
FastGAN (Liu et al., 2021)	0.357	0.370	0.339	0.467	0.430	0.357
MixDL (Kong et al., 2022)	0.296	0.276	0.264	0.305	0.274	0.221
Adaptive IMLE (Ours)	**0.058**	**0.036**	**0.039**	**0.074**	**0.072**	**0.014**

method achieves a better LPIPS backtracking score across all datasets, showing better mode coverage. Due to limited computational resources, we omit the calculation of this metric for EDM (Karras et al., 2022), as it necessitates optimizing a substantial number of noise samples. These results show that our method could produce high-quality images while obtaining better mode coverage compared to the baselines, thereby setting a new state-of-the-art.

Figure 4: Visualizations of the reconstructions of an unseen target image from LPIPS backtracking. While the reconstructions of FakeCLR, MixDL and FastGAN are structurally dissimilar from the target images, the reconstructions of our method are structurally similar to the target images.

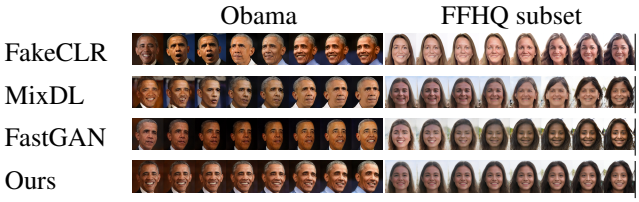

Figure 5: Latent space interpolation results. Our results show smooth and meaningful transitions and higher quality images generated from intermediate points along the interpolation line in the latent space. Start and end of interpolations are nearest neighbours of the same data examples among 200 samples generated by each method.

5.2. Qualitative Results

We show the qualitative comparison of our method to the baselines in Fig. 3. As shown, our method generates higher quality samples which better preserve the semantic structures compared to the baselines, such as the eyes in Cat, the facial structure in Dog and the mouth and hair in the FFHQ subset. In addition, our method generates more diverse results while the GAN-based baselines suffer from mode collapse and generate similar samples, such as in Panda, Grumpy Cat and Obama. Additional samples from our model can be found in Appendix C.

We show the final reconstruction of the target image found using LPIPS backtracking (Liu et al., 2021) on the models trained with different methods in Fig. 4. As shown, our method is the only one where the reconstruction is structurally similar to the target image, demonstrating that our model successfully covers the mode that the target image belongs to.

We also compare our method to the baselines on the quality of intverpolations between two samples in the latent space. As shown in Fig. 5, our method interpolates more smoothly and naturally than the baselines, thereby indicating that our model is less overfitted to the training examples.

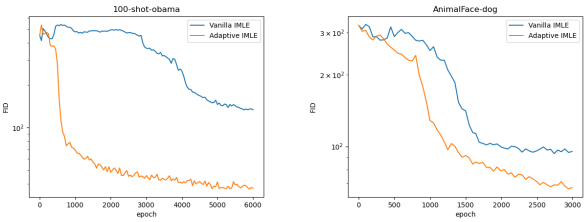

Figure 6: FID comparison of Adaptive IMLE and Vanilla IMLE during training on the Obama and Dog datasets. Adaptive IMLE achieves better image quality (lower FID) and faster convergence rate compared to vanilla IMLE.

Table 4: FID comparison between vanilla IMLE and the proposed method, Adaptive IMLE. Adaptive IMLE consistently outperforms vanilla IMLE, demonstrating better generated image quality.

	Grumpy Cat	Obama	Panda	Cat	Dog	FFHQ subset
	FID↓	FID↓	FID↓	FID↓	FID↓	FID↓
Vanilla IMLE	23.3	37.4	8.2	34.4	61.9	54.1
Adaptive IMLE (Ours)	**19.1**	**25.0**	**7.6**	**24.9**	**43.0**	**33.2**

5.3. Ablation Study

We compare the FID for the proposed method, namely Adaptive IMLE, and vanilla IMLE on all datasets in Tab. 4. The results show that Adaptive IMLE significantly improves upon vanilla IMLE in terms of FID. We also compare the FID scores throughout the training process on Obama and Dog datasets in Fig. 6. The results demonstrate the improvements achieved by Adaptive IMLE in terms of generated image quality and convergence speed, validating the effectiveness of our proposed method in the few-shot setting.

6. Conclusion

We developed a method for the challenging few-shot image synthesis setting that does not depend on pre-training on auxiliary datasets. We presented a more generalized formulation of IMLE and proved that the theoretical guarantees of this generalized formulation hold under weaker conditions. We further derived a novel algorithm based on this formulation which can adapt to different training examples of varying difficulty. We showed that our method significantly outperforms existing baselines in terms of image quality and mode coverage on six few-shot benchmark datasets.

7. Acknowledgements

This research was enabled in part by support provided by NSERC, the BC DRI Group and the Digital Research Alliance of Canada.

References

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *ArXiv*, abs/1809.11096, 2019.

Child, R. Very deep vaes generalize autoregressive models and can outperform them on images. *ArXiv*, abs/2011.10650, 2021.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *ArXiv*, abs/2105.05233, 2021.

Dinh, L., Sohl-Dickstein, J. N., and Bengio, S. Density estimation using real nvp. *ArXiv*, abs/1605.08803, 2017.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/finn17a.html.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of stylegan. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8107–8116, 2020.

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., and Aila, T. Alias-free generative adversarial networks. In *NeurIPS*, 2021.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. *ArXiv*, abs/1807.03039, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kobyzev, I., Prince, S., and Brubaker, M. A. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:3964–3979, 2021.

Kong, C., Kim, J., Han, D., and Kwak, N. Smoothing the generative latent space with mixup-based distance learning. In *European Conference on Computer Vision*, 2022.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

Li, K. and Malik, J. Implicit maximum likelihood estimation. *arXiv preprint arXiv:1809.09087*, 2018.

Li*, K., Peng*, S., Zhang*, T., and Malik, J. Multimodal image synthesis with conditional implicit maximum likelihood estimation. *International Journal of Computer Vision*, May 2020. ISSN 1573-1405. doi: 10.1007/s11263-020-01325-y. URL https://doi.org/10.1007/s11263-020-01325-y.

Li, Y., Zhang, R., Lu, J., and Shechtman, E. Few-shot image generation with elastic weight consolidation. *ArXiv*, abs/2012.02780, 2020.

Li, Z., Wang, C., Zheng, H., Zhang, J., and Li, B. Fakeclr: Exploring contrastive learning for solving latent discontinuity in data-efficient gans. In *ECCV*, 2022.

Liu, B., Zhu, Y., Song, K., and Elgammal, A. Towards faster and stabilized GAN training for high-fidelity few-shot image synthesis. *CoRR*, abs/2101.04775, 2021. URL https://arxiv.org/abs/2101.04775.

Liu, M.-Y., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., and Kautz, J. Few-shot unsupervised image-to-image translation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10550–10559, 2019.

Mo, S., Cho, M., and Shin, J. Freeze discriminator: A simple baseline for fine-tuning gans. *ArXiv*, abs/2002.10964, 2020.

Ojha, U., Li, Y., Lu, J., Efros, A. A., Lee, Y. J., Shechtman, E., and Zhang, R. Few-shot image generation via cross-domain correspondence. *2021 IEEE/CVF Conference on*

Computer Vision and Pattern Recognition (CVPR), pp. 10738–10747, 2021.

O'Mahony, N., Campbell, S., Carvalho, A., Krpalkova, L., Hernandez, G. V., Harapanahalli, S., Riordan, D., and Walsh, J. One-shot learning for custom identification tasks; a review. *Procedia Manufacturing*, 38:186–193, 2019. ISSN 2351-9789. doi: https://doi.org/10.1016/j.promfg.2020.01.025. URL https://www.sciencedirect.com/science/article/pii/S2351978920300263. 29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM 2019), June 24-28, 2019, Limerick, Ireland, Beyond Industry 4.0: Industrial Advances, Engineering Education and Intelligent Manufacturing.

Peng, S., Moazenipourasil, S. A., and Li, K. Chimle: Conditional hierarchical imle. In *NeurIPS*, 2022.

Razavi, A., van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with vq-vae-2. *ArXiv*, abs/1906.00446, 2019.

Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *ArXiv*, abs/1701.05517, 2017.

Si, Z. and Zhu, S.-C. Learning hybrid image templates (hit) by information projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:1354–1367, 2012.

Sinha, A., Song, J., Meng, C., and Ermon, S. D2c: Diffusion-denoising models for few-shot conditional generation. *ArXiv*, abs/2106.06819, 2021.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/cb8da6767461f2812ae4290eac7cbc42-Paper.pdf.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *ArXiv*, abs/1907.05600, 2019.

Song, Y., Sohl-Dickstein, J. N., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *ArXiv*, abs/2011.13456, 2021.

Vahdat, A. and Kautz, J. Nvae: A deep hierarchical variational autoencoder. *ArXiv*, abs/2007.03898, 2020.

van den Oord, A., Kalchbrenner, N., Espeholt, L., Kavukcuoglu, K., Vinyals, O., and Graves, A. Conditional image generation with pixelcnn decoders. *ArXiv*, abs/1606.05328, 2016a.

van den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. *ArXiv*, abs/1601.06759, 2016b.

Wang, T.-C., Liu, M.-Y., Tao, A., Liu, G., Kautz, J., and Catanzaro, B. Few-shot video-to-video synthesis. *ArXiv*, abs/1910.12713, 2019.

Wang, Y., Gonzalez-Garcia, A., Berga, D., Herranz, L., Khan, F. S., and van de Weijer, J. Minegan: Effective knowledge transfer from gans to target domains with few images. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9329–9338, 2020.

Zhao, M., Cong, Y., and Carin, L. On leveraging pretrained gans for generation with limited data. In *ICML*, 2020a.

Zhao, S., Liu, Z., Lin, J., Zhu, J.-Y., and Han, S. Differentiable augmentation for data-efficient gan training. *ArXiv*, abs/2006.10738, 2020b.

A. Proofs

Lemma 1. *Let X be a non-negative random variable and Φ be a continuous function on $[0,\infty)$. If Φ' is integrable on all closed intervals in $[0,\infty)$,*

$$\mathbb{E}\left[\Phi(X)\right] = \Phi(0) + \int_0^\infty \Phi'(t)\mathrm{Pr}(X \geq t)dt$$

Proof.

$$\begin{aligned}
\Phi(0) + \int_0^\infty \Phi'(t)\mathrm{Pr}(X \geq t)dt &= \Phi(0) + \int_0^\infty \int_t^\infty \Phi'(t)p(x)dxdt \\
&= \Phi(0) + \int_{\{x\geq t, t\geq 0\}} \Phi'(t)p(x)d\begin{pmatrix} x \\ t \end{pmatrix} \\
&= \Phi(0) + \int_{\{t\leq x, t\geq 0\}} \Phi'(t)p(x)d\begin{pmatrix} x \\ t \end{pmatrix} \\
&= \Phi(0) + \int_0^\infty \int_0^x \Phi'(t)p(x)dtdx \\
&= \Phi(0) + \int_0^\infty \left(\int_0^x \Phi'(t)dt\right)p(x)dx \\
&= \Phi(0) + \int_0^\infty \left(\Phi(x) - \Phi(0)\right)p(x)dx \quad \text{(2nd FTC)} \\
&= \Phi(0) + \int_0^\infty \Phi(x)p(x)dx - \int_0^\infty \Phi(0)p(x)dx \\
&= \Phi(0) + \int_0^\infty \Phi(x)p(x)dx - \Phi(0)\int_0^\infty p(x)dx \\
&= \Phi(0) + \mathbb{E}\left[\Phi(X)\right] - \Phi(0) \\
&= \mathbb{E}\left[\Phi(X)\right]
\end{aligned}$$

□

Lemma 2. *Under the choice of $\Phi_\tau(\cdot)$ above and its associated $\Phi'_\tau(\cdot)$,*

$$\mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\Phi_{\tau_i}(d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)))\middle|\{\tau_i\}_i\right] = \tau_i - \int_{\delta\tau_i}^{\tau_i} \mathrm{Pr}(d(\mathbf{x}_i, T_\theta(\mathbf{z})) < t)dt.$$

Proof. By definition, $\Phi_{\tau_i}(0) = \delta\tau_i$.

$$\begin{aligned}
\mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\Phi_{\tau_i}(d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)))\middle|\{\tau_i\}_i\right] &= \Phi_{\tau_i}(0) + \int_0^\infty \Phi'_{\tau_i}(t)\mathrm{Pr}(d(\mathbf{x}_i, T_\theta(\mathbf{z})) \geq t)dt \quad \text{(Lemma 1)} \\
&= \delta\tau_i + \int_{\delta\tau_i}^{\tau_i} \mathrm{Pr}(d(\mathbf{x}_i, T_\theta(\mathbf{z})) \geq t)dt \\
&= \delta\tau_i + \int_{\delta\tau_i}^{\tau_i} \left(1 - \mathrm{Pr}(d(\mathbf{x}_i, T_\theta(\mathbf{z})) < t)\right)dt \\
&= \delta\tau_i + (\tau_i - \delta\tau_i) - \int_{\delta\tau_i}^{\tau_i} \mathrm{Pr}(d(\mathbf{x}_i, T_\theta(\mathbf{z})) < t)dt \\
&= \tau_i - \int_{\delta\tau_i}^{\tau_i} \mathrm{Pr}(d(\mathbf{x}_i, T_\theta(\mathbf{z})) < t)dt
\end{aligned}$$

□

Lemma 3. *Under the choice of $\Phi_\tau(\cdot)$ above and its associated $\Phi'_\tau(\cdot)$,*

$$\mathcal{L}_{\{\tau_i\}_i}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{mw_i}\sum_{j=1}^{m}\int_{\delta\tau_i}^{\tau_i}\Pr(d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)) < t)dt.$$

Proof.

$$\begin{aligned}
\mathcal{L}_{\{\tau_i\}_i}(\theta) &= \mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\frac{1}{n}\sum_{i=1}^{n}\frac{1}{w_i}\left(\tau_i - \frac{1}{m}\sum_{j=1}^{m}\Phi_{\tau_i}(d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)))\right)\middle|\,\{\tau_i\}_i\right] \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{w_i}\left(\tau_i - \frac{1}{m}\sum_{j=1}^{m}\mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\Phi_{\tau_i}(d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)))\middle|\,\{\tau_i\}_i\right]\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{w_i}\left(\tau_i - \frac{1}{m}\sum_{j=1}^{m}\left(\tau_i - \int_{\delta\tau_i}^{\tau_i}\Pr(d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)) < t)dt\right)\right) \quad \text{(Lemma 2)} \\
&= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{mw_i}\sum_{j=1}^{m}\int_{\delta\tau_i}^{\tau_i}\Pr(d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)) < t)dt
\end{aligned}$$

□

Equation 3.

Proof.

$$\begin{aligned}
\mathcal{L}_{\{\tau_i\}_i}(\theta) &= \mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\frac{1}{n}\sum_{i=1}^{n}\frac{1}{w_i}\left(\tau_i - \frac{m-1}{m}\tau_i - \frac{1}{m}\max(\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)), \delta\tau_i)\right)\middle|\,\{\tau_i\}_i\right] \\
&= \mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\frac{1}{n}\sum_{i=1}^{n}\frac{1}{w_i}\left(\frac{1}{m}\tau_i - \frac{1}{m}\max(\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)), \delta\tau_i)\right)\middle|\,\{\tau_i\}_i\right] \\
&= \mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\frac{1}{nm}\sum_{i=1}^{n}\frac{1}{w_i}\left(\tau_i - \max(\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)), \delta\tau_i)\right)\middle|\,\{\tau_i\}_i\right]
\end{aligned}$$

□

Equation 4.

Proof.

$$\begin{aligned}
\arg\max_\theta \mathcal{L}_{\{\tau_i\}_i}(\theta) &= \arg\max_\theta \mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\frac{1}{nm}\sum_{i=1}^{n}\frac{1}{w_i}\left(\tau_i - \max(\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)), \delta\tau_i)\right)\middle|\,\{\tau_i\}_i\right] \\
&= \arg\max_\theta \mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\sum_{i=1}^{n}\frac{1}{w_i}\left(\tau_i - \max(\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)), \delta\tau_i)\right)\middle|\,\{\tau_i\}_i\right] \\
&= \arg\max_\theta \mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\sum_{i=1}^{n}\frac{\tau_i}{w_i} - \sum_{i=1}^{n}\frac{1}{w_i}\max(\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)), \delta\tau_i)\middle|\,\{\tau_i\}_i\right] \\
&= \arg\max_\theta \mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[-\sum_{i=1}^{n}\frac{1}{w_i}\max(\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)), \delta\tau_i)\middle|\,\{\tau_i\}_i\right] \\
&= \arg\min_\theta \mathbb{E}_{z_1,\ldots,z_m\sim\mathcal{N}(0,I)}\left[\sum_{i=1}^{n}\frac{1}{w_i}\max(\min_{j\in[m]} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)), \delta\tau_i)\middle|\,\{\tau_i\}_i\right]
\end{aligned}$$

□

Lemma 4. *Suppose p_θ is continuous at all data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, under the choice of $w_i = \int_{\delta\tau_i}^{\tau_i} \mathrm{vol}(B_t(\mathbf{x}_i))dt := \int_{\delta\tau_i}^{\tau_i} \int_{B_t(\mathbf{x}_i)} d\mathbf{x}dt$, where $B_r(\mathbf{x}) = \{\mathbf{y} | d(\mathbf{y}, \mathbf{x}) < r\}$ is an open ball of radius r centred at $\mathbf{x}$,*

$$\lim_{\{\tau_i \to 0^+\}_i} \mathcal{L}_{\{\tau_i\}_i}(\theta) = \frac{1}{n} \sum_{i=1}^{n} p_\theta(\mathbf{x}_i)$$

Proof.

$$\begin{aligned}
\mathcal{L}_{\{\tau_i\}_i}(\theta) &= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m w_i} \sum_{j=1}^{m} \int_{\delta\tau_i}^{\tau_i} \Pr(d(\mathbf{x}_i, T_\theta(\mathbf{z}_j)) < t) dt \quad \text{(Lemma 3)} \\
&= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m w_i} \sum_{j=1}^{m} \int_{\delta\tau_i}^{\tau_i} \int_{B_t(\mathbf{x}_i)} p_\theta(\mathbf{x}) d\mathbf{x} dt \\
&= \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{1}{w_i} \int_{\delta\tau_i}^{\tau_i} \int_{B_t(\mathbf{x}_i)} p_\theta(\mathbf{x}) d\mathbf{x} dt \\
&= \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{\int_{\delta\tau_i}^{\tau_i} \int_{B_t(\mathbf{x}_i)} p_\theta(\mathbf{x}) d\mathbf{x} dt}{\int_{\delta\tau_i}^{\tau_i} \int_{B_t(\mathbf{x}_i)} d\mathbf{x} dt}
\end{aligned}$$

$$\begin{aligned}
\lim_{\{\tau_i \to 0^+\}_i} \mathcal{L}_{\{\tau_i\}_i}(\theta) &= \frac{1}{nm} \sum_{i=1}^{n} \left(\lim_{\tau_i \to 0^+} \left(\sum_{j=1}^{m} \frac{\int_{\delta\tau_i}^{\tau_i} \int_{B_t(\mathbf{x}_i)} p_\theta(\mathbf{x}) d\mathbf{x} dt}{\int_{\delta\tau_i}^{\tau_i} \int_{B_t(\mathbf{x}_i)} d\mathbf{x} dt} \right) \right) \\
&= \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \left(\lim_{\tau_i \to 0^+} \frac{\int_{\delta\tau_i}^{\tau_i} \int_{B_t(\mathbf{x}_i)} p_\theta(\mathbf{x}) d\mathbf{x} dt}{\int_{\delta\tau_i}^{\tau_i} \int_{B_t(\mathbf{x}_i)} d\mathbf{x} dt} \right) \\
&= \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \left(\lim_{\tau_i \to 0^+} \frac{\int_{B_{\tau_i}(\mathbf{x}_i)} p_\theta(\mathbf{x}) d\mathbf{x} - \delta \int_{B_{\delta\tau_i}(\mathbf{x}_i)} p_\theta(\mathbf{x}) d\mathbf{x}}{\int_{B_{\tau_i}(\mathbf{x}_i)} d\mathbf{x} - \delta \int_{B_{\delta\tau_i}(\mathbf{x}_i)} d\mathbf{x}} \right) \quad \text{(L'Hôpital and 2nd FTC)} \\
&= \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \left(\lim_{\tau_i \to 0^+} \frac{\int_{B_{\tau_i}(\mathbf{x}_i)} p_\theta(\mathbf{x}) (1 - \delta \mathbf{1}_{B_{\delta\tau_i}(\mathbf{x}_i)}(\mathbf{x})) d\mathbf{x}}{\int_{B_{\tau_i}(\mathbf{x}_i)} 1 - \delta \mathbf{1}_{B_{\delta\tau_i}(\mathbf{x}_i)}(\mathbf{x}) d\mathbf{x}} \right) \\
&= \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \left(\lim_{\tau_i \to 0^+} \frac{\int_0^{\tau_i} (1 - \delta \mathbf{1}_{\{r < \delta\tau_i\}}(r)) \int_{\{\mathbf{x} | d(\mathbf{x}, \mathbf{x}_i) = r\}} p_\theta(\mathbf{x}) d\mathbf{x} dr}{\int_0^{\tau_i} (1 - \delta \mathbf{1}_{\{r < \delta\tau_i\}}(r)) \int_{\{\mathbf{x} | d(\mathbf{x}, \mathbf{x}_i) = r\}} d\mathbf{x} dr} \right) \\
&= \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \left(\lim_{\tau_i \to 0^+} \frac{\int_{\{\mathbf{x} | d(\mathbf{x}, \mathbf{x}_i) = \tau_i\}} p_\theta(\mathbf{x}) d\mathbf{x}}{\int_{\{\mathbf{x} | d(\mathbf{x}, \mathbf{x}_i) = \tau_i\}} d\mathbf{x}} \right) \quad \text{(L'Hôpital and 2nd FTC)} \\
&= \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} p_\theta(\mathbf{x}_i) \quad \text{(Continuity of } p_\theta\text{)} \\
&= \frac{1}{n} \sum_{i=1}^{n} p_\theta(\mathbf{x}_i)
\end{aligned}$$

□

Note that under common metrics like ℓ_p distances, w_i can be found in closed form, i.e., $\mathrm{vol}(B_t(\mathbf{x}_i)) = (2t)^d \frac{\Gamma(1+1/p)^d}{\Gamma(1+d/p)}$, and so $w_i = \int_{\delta\tau_i}^{\tau_i} \mathrm{vol}(B_t(\mathbf{x}_i))dt = \int_{\delta\tau_i}^{\tau_i} (2t)^d \frac{\Gamma(1+1/p)^d}{\Gamma(1+d/p)} dt = \frac{(2(1-\delta)\tau_i)^{d+1}}{2(d+1)} \cdot \frac{\Gamma(1+1/p)^d}{\Gamma(1+d/p)}$, where $\Gamma(\cdot)$ denotes the gamma function.

B. Adaptive IMLE Algorithm In Practice

The Adaptive IMLE algorithm solves a sequence of optimization problems for each data example x_i as described in 4.2. Once an optimization problem is sovled for x_i, i.e., its nearest neighbour gets within the threshold of the ball whose radii is $\delta\tau_i$, a fresh nearest neighbour needs to be assigned to x_i in order to start solving the next optimization problem. This requires generating a set of random samples. Because sample generation is expensive and could become the bottleneck for the algorithm's speed, we would like to reduce it to have an efficient and practical algorithm. To achieve this, we utilize two ideas simultaneously, 1- we store the pool of generated samples and share it among all data examples. This approach enables us to generate a larger sample pool size. As a result, each data example has a higher likelihood of finding a nearby nearest neighbour, thereby leading to faster convergence. 2- we try to reuse this sample pool as long as possible. To achieve the latter, we regenerate a new pool of samples only if a data example x_i needs a new nearest neighbour while having already accessed the sample pool, as indicated by the variable sample_pool_accessed in the algorithm. In rare cases where the sample pool doesn't get updated for a large number of iterations, denoted by t (maximum sample pool age), we regenerate a new pool of samples.

Algorithm 3 Adaptive IMLE Procedure

Require: The set of inputs $\{\mathbf{x}_i\}_{i=1}^n$, tightening coefficient $\delta \in [0, 1)$, maximum pool age $t \in \mathbb{N}$
Initialize the parameters θ of the generator T_θ
Draw latent codes $Z \leftarrow \mathbf{z}_1, ..., \mathbf{z}_m$ from $\mathcal{N}(0, \mathbf{I})$
$\mathbf{z}_i^* \leftarrow \arg\min_{\mathbf{z}_j \in Z} d(\mathbf{x}_i, T_\theta(\mathbf{z}_j))$
$\tau_i \leftarrow d\left(\mathbf{x}_i, T_\theta\left(\mathbf{z}_i^*\right)\right) \ \forall i \in [n]$
sample_pool $\leftarrow \{\}$
sample_pool_age $\leftarrow t$ {number of epochs that the generated samples pool hasn't been updated}
sample_pool_accessed $\leftarrow \{false\}_{i=1}^n$ {keeps track of data examples that have used the sample_pool}
for $k = 1$ **to** K **do**
  Pick a random batch $S \subseteq [n]$
  $S' \leftarrow \{i | d\left(\mathbf{x}_i, T_\theta\left(\mathbf{z}_i^*\right)\right) \leq \delta\tau_i \quad \forall i \in S\}$ {set of data examples to get updated with a fresh nearest neighbour}
  if sample_pool_age $\geq t$ **or** $true \in \{\text{sample\_pool\_accessed}_i \quad | \quad \forall i \in S\}$ **then**
    {we get here if the sample_pool is too old or some data example in S' has used it already}
    Draw latent codes $Z \leftarrow \mathbf{z}_1, ..., \mathbf{z}_m$ from $\mathcal{N}(0, \mathbf{I})$
    sample_pool $\leftarrow \{T_\theta(\mathbf{z}_i) \quad | \quad \forall \mathbf{z}_i \in Z\}$
    sample_pool_accessed $\leftarrow \{false\}_{i=1}^n$ {No data example has used these set of new samples}
    sample_pool_age $\leftarrow 0$
  end if
  for $i \in S'$ **do**
    $j \leftarrow \text{idxmin}_{\mathbf{x}'_j \in \text{sample\_pool}} d(\mathbf{x}_i, \mathbf{x}'_j)$
    $\mathbf{z}_i^* \leftarrow Z_j$
    $\tau_i \leftarrow d(\mathbf{x}_i, T_\theta(\mathbf{z}_i^*))$ {Update the threshold}
    $\text{sample\_pool\_accessed}_i \leftarrow true$
  end for
  $\theta \leftarrow \theta - \eta\nabla_\theta \left(\sum_{i \in S} d\left(\mathbf{x}_i, T_\theta\left(\mathbf{z}_i^*\right)\right)\right) / |S|$
  sample_pool_age $\leftarrow$ sample_pool_age $+ 1$
end for
return θ

C. Additional Results

We show more interpolation results for Adaptive IMLE on FFHQ subset and Obama in Fig. 7. We also show randomly generated samples for FFHQ subset in Fig. 8. For more results, please refer to https://github.com/mehranagh20/AdaIMLE.

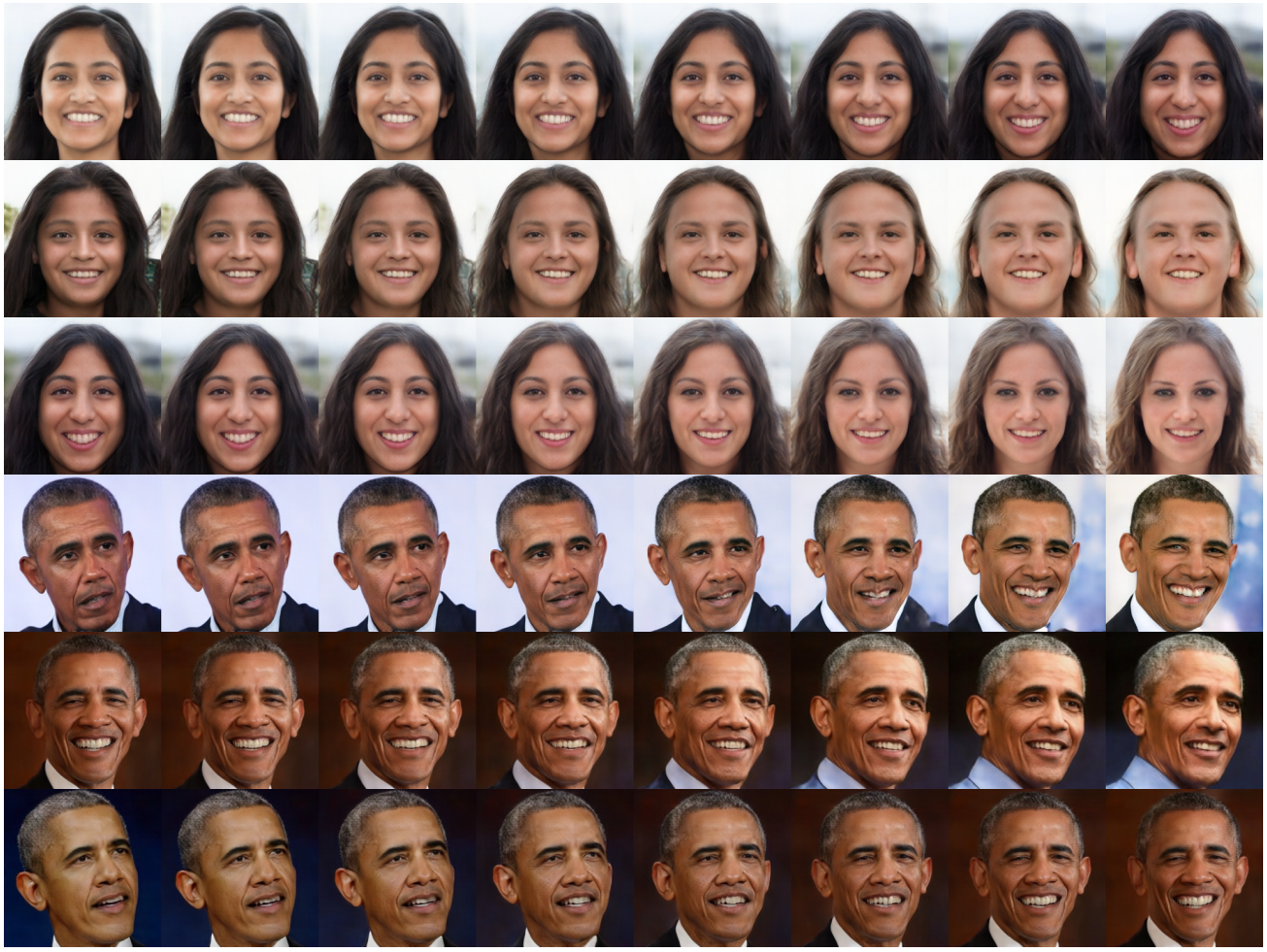

Figure 7: Interpolation results for Adaptive IMLE (Ours). Each row shows a different interpolation.

Figure 8: Adaptive IMLE (Ours): randomly generated samples for *FFHQ subset*.