Principled Long-Tailed Generative Modeling via Diffusion Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Deep generative models, particularly diffusion models, have achieved remarkable success across diverse domains but face significant challenges when trained on real-world, long-tailed datasets-where a few "head" classes dominate and many "tail" classes are underrepresented. This paper develops a rigorous theoretical framework for long-tailed learning via diffusion models through the lens of deep mutual learning. We introduce a novel regularized training objective that combines the standard diffusion loss with a mutual learning term, enabling balanced performance across all class labels, including the underrepresented tails. Our approach to learn via the proposed regularized objective is to formulated it as a multi-player game, with Nash equilibrium serving as the solution concept. We derive a non-asymptotic first-order convergence result for individual gradient descent algorithm to find the Nash equilibrium. We show that the Nash gap of the score network obtained from the algorithm is upper bounded by $\mathcal{O}(\frac{1}{\sqrt{T_{train}}} + \beta)$ where β is the regularizing parameter and T_{train} is the number of iterations of the training algorithm. Furthermore, we theoretically establish hyper-parameters for training and sampling algorithm that ensure we find conditional score networks (under our model) with a worst case sampling error $\mathcal{O}(\epsilon + 1), \forall \epsilon > 0$ across all class labels, which is further supported by numerical experiments. Our results offer new insights and guarantees for training diffusion models on imbalanced, long-tailed data, with implications for fairness, privacy, and generalization in real-world generative modeling scenarios.

1 Introduction

Successful integration of deep learning models into society requires working with real-world data. This comes with many challenges: data quality issues such as inaccurate data, data bias, ethical issues such as breach in privacy, transparency, technical issues such as data integration, generalization, scalability, etc. Furthermore, real world class-labeled datasets are not uniform, but follow a skewed or sometimes referred to as “long-tailed" distribution. It is characterized by a “head" classes that occurs with high probability while the probability of the rest of the classes, often referred to as “tail" classes falls off very quickly. It is well known that the performance of traditional deep learning ([12, 16]) and generative models ([28]) suffer significantly when trained on *long-tailed* distributions.

One might be curious to ask, *“Should deep learning or generative models be concerned with class labels which occur with very low frequency?"*

The answer is Yes!!! Even though individually the classes occur with low frequency, collectively they represent a significant portion of classes and collectively occur with high probability. Diffusion models, which are the focus of this work are no exception to this phenomenon. Diffusion models are latent variable generative models which learn diffusion process for a given dataset, such that the process can generate new elements that are distributed similarly as the original dataset (See section 3 for more details). They have become popular techniques in image generation (beating traditional models such as GANs [5]) [2, 35], natural language processing [36], time series forecasting [24] and in fields of applied chemistry [1], biology [7] and medicine [13] to name a few. However, the study of diffusion models for long-tailed learning is limited. [21] showed that when the traditional conditional Diffusion Denoising Probabilistic Model (DDPM) is trained on a long-tailed distribution, the conditional DDPM model as shown in [21, Figure 1], "generates head class images with satisfying performance, whereas conversely, the generated images on tail classes are very likely to show unrecognizable semantics". Moreover, there might be privacy and ethical concerns if the model overfit (memorize) to the tail class label data and replicate them during generation.

Motivated by this, the current work develops the theory of *Long-Tailed Learning* for diffusion models in a mathematically rigorous manner through the perspective of *Deep Mutual Learning*. Our main results include the following:

1. Under a suitable metric (KL- divergence) that captures the distance between the learnt class distribution and the ground truth class distribution, we derive an upper bound on the worst case distance across all class labels. To do so, we employ Deep Mutual Learning along with the usual score based diffusion model objective in literature [27, 29]. We present the formulation as a game across conditional score networks and propose Nash equilibrium of the game as the egalitarian solution concept.

2. Borrowing ideas from [11] on Deep Mutual Learning, we derive a non-asymptotic first order convergence result for the individual gradient descent algorithm to find the Nash Equilibrium of the proposed game. We show the Nash gap of the score network obtained from the algorithm is upper bounded by $\mathcal{O}(\frac{1}{\sqrt{T_{train}}} + \beta)$ where β is the regularizing parameter and T_{train} is the number of iterations of the training algorithm. Finally, we show we can find hyper-parameters for training and sampling such that the score networks obtained from the algorithm enjoys a worst case error bound of $\mathcal{O}(\epsilon + 1)$ for any $\epsilon > 0$ for any class, tail and otherwise.

2 Related Works

Long-Tailed Learning for Diffusion Models. To tackle the issue with *long-tailed* distributions, diverse techniques that can be adopted at various phases of implementation of deep learning models such as re-sampling [25], re-weighting [25], transfer learning [21], and feature augmentation [8] have been proposed. The closest related work to ours is that of [21] titled "Class- Balancing Diffusion Models" or CBDM and its followups [30, 32]. The paper proposes a distribution adjustment regularizer as a solution along with the usual DDPM objective. This represents a modification in the training phase. Their experiments show that the images generated by CBDM exhibit greater diversity and quality in both quantitative and qualitative ways when trained on CIFAR100/CIFAR100LT datasets. As mentioned in [30], *"CBDM [21] represents an inaugural inquiry into the performance of DDPM within the context of long-tailed data scenarios"*. Motivated by CBDM and contrastive learning, [30] propose adding a penalty function to demarcate the distribution boundaries of different data categories. However, the derivation of the distribution adjustment regularizer in [21, 22, Proposition 2, Appendix A] rely on strong assumptions. They follow a traditional machine learning framework that optimizes over a single objective function with a single neural network and give empirical verification of their method's performance. On the other hand, we define a game across conditional score networks and propose the Nash equilibrium of this game as the egalitarian solution to learn a fair score function for equally good generation over all classes. Furthermore, our framework and analysis do not rely on the strong assumptions made in [21, 22, Proposition 2, Appendix A].

Deep Mutual Learning. Deep Mutual Learning (DML) [33] is a knowledge distillation process that allows the transfer of knowledge from a highly powerful model to a smaller faster efficient model. In DML, an ensemble of students (models) learn collaboratively and teach each other throughout the training process. DML has shown promise in visual object tracking [34], metric learning [20], multi-modal recommender systems [14], and classification tasks trained on *Long-Tailed* distributions [19]. The theoretical performance guarantees for models trained with DML is scarce. [11] gives a non-asymptotic first order convergence result for training models for classification task using DML. Deep Mutual Learning literature propose various methods for optimizing Deep Mutual Learning objectives without specifying the solution concept they seek. In contrast, we show that the individual gradient descent (one method for DML) is seeking a Nash equilibrium of an underling multiplayer game. While this result of our could be of independent interest, in this work we further leverage this result to obtain a generalization result for diffusion models for long-tailed generation.

Training and Sampling of Score-based (Conditional) Diffusion Models. The performance of score-based diffusion models have been rigorously studied in the literature of generative modeling. [9, 15, 29] provided a full error analysis of training and sampling from a diffusion model. [9, 15] parametrize the score network using a random feature model and use gradient flow to train the model. [9] leverages Neural Tangent Kernels to obtain an approximation and generalization error for diffusion models. [29] parametrize the score network by a deep neural network and prove exponential convergence of its gradient descent training dynamic on the empirical loss function. For conditional diffusion models, [6] provides data- dependent approximation bounds of the conditional score function by multi-layered neural network and also give an expected sampling error of the approximated distribution over all class labels. Compared to [6, 29], we consider a finite label class and make no assumption on how the data is distributed. While our result can readily be extended to deeper neural network in line with [29], we parametrize the score function using a two-layer ReLU network (as in [9, 15]) due to the nice properties it induces in the proposed game.

3 Basics of Score-Based Diffsuion Generative Models

Notations. Let $\|.\|$ denote the ℓ_2 norm for vectors and matrices, $\|.\|_F$ be the Frobenius norm. For the discrete time points, we use t_i to denote time point for forward dynamics and $t_i^{\leftarrow}$ for backward dynamics. $\sigma(x)$ where $x \in \mathbb{R}^d$ refers to the ReLU activation function applied element-wise while $\bar{\sigma}_t$ refers to the variance of the forward dynamics. $\tau \in [T_{train}]$ represents the iteration of the training algorithm, which in our case is gradient descent. θ_y is the training parameter for score for label $y \in \mathcal{Y}$ while θ_{-y} is the training parameter for score for all label $y' \in \mathcal{Y} - \{y\}$.

In subsequent sections, we introduce the basics of diffusion model training and generation. Denote the initial conditional distribution as $P_0(X_0 = x|y), \forall x \in \mathcal{A} \subset \mathbb{R}^d$, $\mathcal{A}$ is a compact set of all possible features and $y \in \mathcal{Y}$ where $\mathcal{Y}$ is the finite set of class labels.

3.1 Forward and Backward Processes

The use of diffusion model in generative modeling involves two processes:

1. **Forward Process:** The forward process that pushes an initial distribution $P_0(.|y)$ to Gaussian by adding noise progressively to X_0. This forward dynamics is usually described as the Ornstein-Uhlenbeck (OU) process,
$$dX_t = -f_t X_t dt + g_t dW_t, \text{ with } X_0 \sim P_0(.|y), \quad \forall y \in \mathcal{Y}, \tag{1}$$
where f_t, g_t are functions of $t \in [0, T]$ and dW_t is the incremental Brownian motion or Wiener process, X_t is a $d-$ dimensional random variable with $X_t \sim P_t(.|y)$. The choice of f_t, g_t results in various diffusion model schemes such as Variance Preserving (VP), Variance Exploding (VE) SDE (See [27] for more details).
2. **Backward Process:** To generate a new sample, the above mentioned forward dynamics can be reversed conditioned on the final distribution $X_T^{\leftarrow} \sim P_0(.|y)$ to get the backward or reverse diffusion process defined as:
$$dX_t^{\leftarrow} = \left[f_{T-t} X_t^{\leftarrow} + g_{T-t}^2 \nabla_x \log p_{T-t}(X_t^{\leftarrow}|y) \right] dt + g_{T-t} d\bar{W}_t, X_0^{\leftarrow} \sim P_T(.|y), \tag{2}$$
where $X_0^{\leftarrow} \sim P_T$ and p_t is the density of P_t. Then $X_{T-t}^{\leftarrow}$ and X_t have the same distribution with density $P_t(.|y)$, which means the dynamics will push near Gaussian distributions back to the initial distribution $P_0(.|y), \forall y \in \mathcal{Y}$.

3.2 Training via Denoising Score Matching

As described in eq. 2, to generate samples conditionally, one needs access to $\nabla_x \log p_{T-t}(X_t^{\leftarrow}|y)$, the conditional score function, which is unknown. Let $s_{t,\theta}(x, y)$ be an estimator of $\nabla_x \log p_t(x|y)$. To estimate the conditional score function, a natural loss function to train a model would be the following objective:
$$\mathcal{L}_{conti}(\theta) = \mathbb{E}_y[\mathcal{L}_{conti}^y(\theta)] = \frac{1}{2} \int_{t_0}^T \lambda(t) \mathbb{E}_{(x(t),y)} \left[\left\| \nabla_{x(t)} \log p_t(x(t)|y) - s_{t,\theta}(x(t), y) \right\|_2^2 \right] dt, \tag{3}$$
where
$$\mathcal{L}_{conti}^y(\theta) = \frac{1}{2} \int_{t_0}^T \lambda(t) \mathbb{E}_{x(t)} \left[\left\| \nabla_{x(t)} \log p_t(x(t)|y) - s_{t,\theta}(x(t), y) \right\|_2^2 \right]. \tag{4}$$

Once the conditional score function is learnt, a datum from class label $y \in \mathcal{Y}$ is sampled using the reverse diffusion process given below:

$$d\tilde{X}_t^{\leftarrow} = \left[f_{T-t}\tilde{X}_t^{\leftarrow} + g_{T-t}^2 s_{t,\theta}(x(T-t), y) \right] dt + g_{T-t} d\bar{W}_t, \text{ with } \tilde{X}_0^{\leftarrow} \sim \mathcal{N}(0, I). \tag{5}$$

To measure how well the learnt score function approximates the ground truth distribution, the KL-divergence is employed as the metric.

Definition 1 (KL Divergence)**.** *Given two distributions p and q, the KL divergence from q to p is defined as $D_{KL}(p||q) = \int_{R^d} p(x) \log \frac{p(x)}{q(x)} dx$.*

To assess the goodness of the learnt score function through the optimization of $\mathcal{L}^y_{conti}(\theta)$, we have to relate the KL-divergence between the learnt distribution and the ground truth to the training objective. Informally, the KL divergence between the learned distribution and the ground truth distribution is bounded by the score based diffusion model objective ($\mathcal{L}_{conti}(\theta)$) as (See [26, Theorem 1] or [6, Appendix D] for detailed proof)

$$\mathbb{E}_y \left[\mathcal{D}_{KL}(P_0(.|y)||P_{0,\theta}(.|y)) \right] \lesssim \mathcal{L}_{conti}(\theta) \tag{6}$$

Achieving a bound on the expectation as [6, Theorem 4.1] gives no insights into the worst case sampling error over all $y \in \mathcal{Y}$. In this work, we provide a methodology to achieve an upper bound on $\max_{y \in \mathcal{Y}} \mathcal{D}_{KL}(P_0(.|y)||P_{0,\theta}(.|y))$, thereby addressing the long-tailed issue in generative modeling.

4 Long-Tailed Learning

4.1 Egalitarian Solution Concept

Previous work in conditional diffusion models [6] have focused on optimizing the following objective

$$\mathcal{L}_{conti}(\theta) = \mathbb{E}_y \left[\mathcal{L}^y_{conti}(\theta) \right] = \sum_{y \in \mathcal{Y}} p(y) \mathcal{L}^y_{conti}(\theta) \tag{7}$$

for classifier guided sampling [27] or the unconditional score function along with the conditional score function from 7 for classifier free guidance. The above objective is sound when the marginal density of the classes $p(y)$ itself is uniformly distributed. Observe that when optimizing $\mathcal{L}_{conti}(\theta)$ (eq. 7), an optimization algorithm will give more weight towards reducing $\mathcal{L}^y(\theta)$ for head classes (classes with high $p(y)$, appearing with higher frequency in the data). Thus, the trained model overfits the head class, while performing poorly on the tail classes. One way of ensuring that each class label is equally weighted during the training process is to re-weigh each class objective function by a factor inversely proportional to the class marginal density $p(y)$. This ensure that both head and tail classes receive equal weighting during the training process.

$$\mathcal{L}_{conti,balanced}(\theta) = \mathbb{E}_y \left[\frac{1}{p(y)} \mathcal{L}^y_{conti}(\theta) \right] = \sum_{y \in \mathcal{Y}} \mathcal{L}^y_{conti}(\theta). \tag{8}$$

However, in many real world scenarios the marginal density $p(y)$ is unknown and hence such an accurate reweighting is not possible. For *Long-Tailed Learning*, as we desire to perform well (in terms of generation quality) for every class label, the natural objective would be to minimize $\max_{y \in \mathcal{Y}} \mathcal{D}_{KL}(P_0(.|y)||P_{0,\theta}(.|y))$, that is, minimize the worst-case KL divergence over all $y \in \mathcal{Y}$. Suppose $\mathcal{L}^y_{conti}(\theta)$ is convex in the training parameter θ, then so is $f(\theta) = \max_{y \in \mathcal{Y}} \mathcal{L}^y_{conti}(\theta)$ as maximum of finite convex functions is again convex. $f(\theta)$ may not be differentiable even if $\mathcal{L}^y_{conti}(\theta)$ are differentiable in θ for all $y \in \mathcal{Y}$. One could use sub-gradient methods to optimize the worst case class loss $\max_y \mathcal{L}^y(\theta)$. However, in practice one has to work with the empirical version of these loses which might be noisy and lead to parameters that are sub-optimal with respect to the population loss.

4.2 Nash Equilibrium as a Solution Concept

To enable diffusion models for *Long-tailed* learning, we modify the DM objective to add the mutual learning objective defined as

$$\mathcal{L}^y_{conti,mut}(\theta_y, \theta_{-y}, \omega(.)) = \frac{1}{2} \int_{t_0}^{T} \omega(t) \mathbb{E}_{x(t) \sim p_t} \mathbb{E}_{y' \sim Q} \left[\left\| s_{t,\theta_y}(x(t)) - s_{t,\theta_{y'}}(x(t)) \right\|_2^2 \right] dt, \tag{9}$$

to obtain a regularized version of the DM objective function $\mathcal{L}^y_{cont,reg}(\theta_y, \theta_{-y})$. In the setting of Mutual Learning, the distribution $\mathcal{Q}$ is uniform. But, the distribution can be a hyperparameter over which one could optimize. From now on, we will drop the weighting arguments $\lambda(.), \omega(.)$ in the objective functions, leading to the following regularized objective for each class:

$$\mathcal{L}^y_{conti,reg}(\theta_y, \theta_{-y}) = \mathcal{L}^y_{conti}(\theta_y) + \beta \mathcal{L}^y_{conti,mut}(\theta_y, \theta_{-y}). \tag{10}$$

Learning the score $\nabla_{x(t)} \log p_t(x(t)|y)$ is difficult as it is intractable. Conditioning on X_0 and using law of iterated expectation, one can rewrite the objective function as (see [29, Appendix A] for detailed proof) with discretized time points as $0 < t_0 < t_1 < \cdots t_N = T$ to get the training objective

$$\begin{aligned}
\mathcal{L}^y_{reg}(\theta_y, \theta_{-y}) &= \mathcal{L}^y(\theta_y) + \beta \mathcal{L}^y_{mut}(\theta_y, \theta_{-y}) \\
&= \frac{1}{2} \sum_{j=1}^{N} \lambda(t_j)(t_j - t_{j-1}) \mathbb{E}_{X_0} \mathbb{E}_{X_{t_j}|X_0} \left[\left\| \nabla_{x(t_j)} \log p_t(x_i(t_j)|x_0) - s_{t_j,\theta_y}(x_i(t_j)) \right\|_2^2 \right] + \\
&+ \bar{C}(y) + \beta \frac{1}{2} \sum_{j=1}^{N} \omega(t_j)(t_j - t_{j-1}) \mathbb{E}_{X_0} \mathbb{E}_{X_{t_j}|X_0} \mathbb{E}_{y' \sim Q} \left[\left\| s_{t_j,\theta_y}(x_i(t_j)) - s_{t,\theta_{y'}}(x_i(t_j)) \right\|_2^2 \right] \Bigg],
\end{aligned} \tag{11}$$

where $\bar{C}(y) = \frac{1}{2} \sum_{j=1}^{N} \lambda(t_j)(t_j - t_{j-1}) C_{t_j}(y)$ and $C_t(y) = \mathbb{E}_{X_t} \|\nabla \log p_t(.|y)\|^2 - \mathbb{E}_{X_0} \mathbb{E}_{X_t|X_0} \|\nabla \log p_t(x_t|x_0, y)\|^2$. [29, Remark 1] point out that $C(y) < 0$ and hence the first summand in Eq. 11 is always bound below by $-C(y)$. $\bar{C}(y)$ along with the entire first summation in 11 correspond to $\mathcal{L}^y(\theta_y)$ while the third term is $\mathcal{L}^y_{mut}(\theta_y, \theta_{-y})$. As $\bar{C}(y)$ doesn't depend on θ, we can ignore it for the purpose of training. But we not that $C(y)$ will appear in our final worst case sampling error. When the drift and diffusion coefficient of the forward dynamics satisfy some nice properties, the distribution of $p_t(x_t|x_0)$ is normally distributed, whose mean and variance ($\bar{\sigma}_t$) can be explicitly computed. Exploiting this knowledge, one can rewrite the objective function in eq 11 as (See Appendix B.3 for details)

$$\begin{aligned}
\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y}) &= \bar{\mathcal{L}}^{n_y}(\theta_y) + \beta \bar{\mathcal{L}}^{n_y}_{mut}(\theta_y, \theta_{-y}) \\
&= \frac{1}{2n_y} \sum_{i=1}^{n_y} \sum_{j=1}^{N} \frac{\lambda(t_j)(t_j - t_{j-1})}{\bar{\sigma}_{t_j}} \Bigg[\left\| \bar{\sigma}_{t_j} s_{t_j,\theta_y}(x_i(t_j)) + \xi_{ij} \right\|_2^2 \\
&\quad + \beta \omega(t_j)(t_j - t_{j-1}) \mathbb{E}_{y' \sim Q} \left[\left\| s_{t_j,\theta_y}(x_i(t_j)) - s_{t,\theta_{y'}}(x_i(t_j)) \right\|_2^2 \right] \Bigg]
\end{aligned} \tag{12}$$

where $\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y})$ is the empirical version of $\mathcal{L}^y_{reg}(\theta_y, \theta_{-y})$ with n_y samples, $\{x_i\}_{i=1}^{n_y}$ with $x_i \sim P_0(.|y)$ denotes the initial data, $\{\xi_{ij}\}_{j=1}^{N}$ where $\xi_{ij} \sim \mathcal{N}(0, I_d)$ denotes the noise and input data of the neural network is $\{t_j, x_i(t_j)\}_{i=1,j=1}^{n_y,N}$, where $x_i(t_j) \sim P_{t_j}(.|y)$ is obtained from the forward diffusion process.

4.2.1 Neural Network Architecture for Score Parametrization

The approximation power of two-layer ReLU network with randomly sampled input layer are well understood from numerous works [10, 23] and has been used to study the generalization properties of Diffusion Models in [9, 15]. We also parametrize the score function s_{t,θ_y} for each label $y \in \mathcal{Y}$ using a random feature model

$$s_{t,\theta_y}(x) := \frac{1}{m} A_y \sigma(W_y x + U_y e(t)) = \frac{1}{m} \sum_{i=1}^{m} a_{y,i} \sigma(w_{y,i}^T x + u_{y,i}^T e(t)) \tag{13}$$

where $\sigma(\cdot) = \max\{0, \cdot\}$ is the ReLu activation function, $A_y = (a_{y,1}, \cdots, a_{y,m}) \in \mathbb{R}^{d \times m}$ is the trainable parameter, $W_y = (w_{y,1}, \cdots, w_{y,m})^T \in \mathbb{R}^{m \times d}$ and $U_y = (u_{y,1} \cdots, u_{y,m})^T \in \mathbb{R}^{d \times d_e}$ are randomly initialized embedding matrices that are frozen during training, $e : \mathbb{R}_{\geq 0} \to \mathbb{R}^{d_e}$ is the embedding function for the time. The above model represents a neural network with one hidden layer with m neurons and a $d-$ dimensional vector as an output. Suppose $a_{y,i}, w_{y,i}$ and $u_{y,i}$ are i.i.d sampled from an underlying distribution ρ. Then as $m \to \infty$, we can view

$$s_{t,\theta_y}(x) \to \bar{s}_{t,\bar{\theta}_y}(x) := \mathbb{E}_{a_y,w_y,u_y} \left[a_y \sigma(w_y^T x + u_y^T e(t)) \right] = \mathbb{E}_{w,u} \left[a_y(w,u) \sigma(w_y^T x + u_y^T e(t)) \right],$$

Algorithm 1 Individual Gradient Descent

Input parameters: Learning rate η_τ
Initialize: $(W_y, U_y)_{y\in\mathcal{Y}}$ and $\theta_y^0, \forall y \in \mathcal{Y}$
for $\tau = 0...T_{train}$ **do**
  for $y = 0...|\mathcal{Y}|$ **do**
    $\theta_y^{\tau+1} \leftarrow \theta_y^\tau - \eta_\tau \nabla_{\theta_y} \bar{\mathcal{L}}_{reg}^{n_y}(\theta_y^\tau, \theta_{-y}^\tau)$
  end for
end for
Output: $(\theta_y, \theta_{-y}) = \min_{\tau\in[T_{train}]}$ NE-gap$(\theta_y^\tau, \theta_{-y}^\tau)$

with $a_y(w,u) := \frac{1}{\rho_0(w,u)} \int_{\mathbb{R}^d} a_y \rho(a,w,u) da_y$ and $\rho_0(w,u) := \int_{\mathbb{R}^d} \rho(a,w,u)da$. The above relation represents $s_{t,\theta_y}(x)$ as an approximation of the continuous version $\bar{s}_{t,\bar{\theta}_y}(x)$, which can be viewed as a neural network with infinite width, i.e., infinite number of neurons in the hidden layer $(m \to \infty)$. Furthermore, we assume the embedding matrices W_y and U_y are sampled independently for every $y \in \mathcal{Y}$ from a set with bounded support.

Having defined our loss function, we define the strategy space as $\Theta_y = \{A_y \in \mathbb{R}^{d\times m} : \|A_y\|_F \leq B\}, \forall y \in \mathcal{Y}$. Now, consider the $|\mathcal{Y}|$-player game $\langle \mathcal{Y}, (\bar{\mathcal{L}}_{reg}^{n_y})_{y\in\mathcal{Y}}, (\Theta_y)_{y\in\mathcal{Y}}\rangle$,

Definition 2 (Nash Gap)**.** *Let $B_y : \Theta_{-y} \to \Theta_y$ represent the best response function for label $y \in \mathcal{Y}$ defined as $B_y(\theta_{-y}) \in \operatorname{argmin}_{\theta\in\Theta_y} \bar{\mathcal{L}}_{reg}^{n_y}(\theta, \theta_{-y})$. Using the best response function, we define the Nash gap of a strategy profile $(\theta_y)_{y\in\mathcal{Y}} \in \times_{y\in\mathcal{Y}}\Theta_y$ as:*

$$\textit{NE-gap}((\theta_y)_{y\in\mathcal{Y}}) = \max_{y\in\mathcal{Y}} \bar{\mathcal{L}}_{reg}^{n_y}(\theta_y, \theta_{-y}) - \bar{\mathcal{L}}_{reg}^{n_y}(B_y(\theta_{-y}), \theta_{-y}). \tag{14}$$

Definition 3 (Nash Equilibrium)**.** *A strategy $(\theta_y^{'})_{y\in\mathcal{Y}} \in \times_{y\in\mathcal{Y}}\Theta_y$ is an ϵ- Nash equilibrium of the game $\langle \mathcal{Y}, (\bar{\mathcal{L}}_{reg}^{n_y})_{y\in\mathcal{Y}}, (\Theta_y)_{y\in\mathcal{Y}}\rangle$ if NE-gap$((\theta_y^{'})_{y\in\mathcal{Y}}) \leq \epsilon$. When NE-gap$((\theta_y^{'})_{y\in\mathcal{Y}}) = 0$, then $(\theta_y^*)_{y\in\mathcal{Y}}$ is a Nash equilibrium.*

The ability to find an ϵ- Nash equilibrium of the game $\langle \mathcal{Y}, (\bar{\mathcal{L}}_{reg}^{n_y})_{y\in\mathcal{Y}}, (\Theta_y)_{y\in\mathcal{Y}}\rangle$ is crucial in our analysis to bound the worst case sampling error.

4.3 Algorithm

In this section, we propose the individual gradient descent algorithm 1 to find an approximate Nash equilibrium of the game $\langle \mathcal{Y}, (\bar{\mathcal{L}}_{reg}^{n_y})_{y\in\mathcal{Y}}, (\Theta_y)_{y\in\mathcal{Y}}\rangle$. The input parameter for the algorithm is the step-size η_τ where τ is the τ^{th} step of the individual gradient descent algorithm. The initialization step samples the embedding matrices and fixes an initial condition for the training parameter $(W_y, U_y, \theta_y^{(0)})_{y\in\mathcal{Y}}$. The individual gradient proceeds for T_{train} steps and within each step an individual gradient update is performed by computing the gradient $\nabla_{\theta_y} \bar{\mathcal{L}}_{reg}^{n_y}(\theta_y^\tau, \theta_{-y}^\tau)$.

The complexity of finding Nash equilibrium: One of the most celebrated results in game theory [4] proved that the computational complexity of the problem of computing of a Nash equilibrium in an arbitrary game lies in the complexity class PPAD. So far, there does not exist an polynomial time algorithm that can find an approximate or exact solution to problems in PPAD. The game $\langle \mathcal{Y}, (\bar{\mathcal{L}}_{reg}^{n_y})_{y\in\mathcal{Y}}, (\Theta_y)_{y\in\mathcal{Y}}\rangle$ is a convex minimization game (See B.5). [18] showed that concave maximization games (convex minimization games) also lie in the class PPAD. We present a positive result that in our game $\langle \mathcal{Y}, (\bar{\mathcal{L}}_{reg}^{n_y})_{y\in\mathcal{Y}}, (\Theta_y)_{y\in\mathcal{Y}}\rangle$, individual gradient descent finds an approximate Nash equilibrium whose NE-gap is bounded by $\mathcal{O}(\frac{m^2}{\sqrt{T_{train}}} + \beta)$.

5 Main Result

We now present the main result of the capability of diffusion models in long-tailed learning through deep mutual learning. We derive a data-independent worst-case bound for $D_{KL}(p_0(.|y)||p_{0,\theta_{y,t}}(.|y))$. Let $\theta_y^* = \operatorname{argmin}_{\theta_y} \mathcal{L}^y(\theta_y), \forall y \in \mathcal{Y}$ and let $\bar{\theta}_y^*$ be the optimal solution when the true score function $s_{t,\theta_y}(x)$ is replaced in the class-label objective function $\mathcal{L}^y(\theta_y)$ (equation 11) by its approximation $\bar{s}_{t,\bar{\theta}_y}(x)$. We make one assumption on the support of data distribution (justified in Remark 1).

Assumption 1. *We assume that the target distribution $P_0(x|y)$ is continuously differentiable in x and has compact support for every $y \in \mathcal{Y}$. Let for any $y \in \mathcal{Y}$, $x \in \mathcal{A} \subset \mathbb{R}^d$, $\|x\|_\infty \leq K$*

Generation Algorithm. We consider the DDPM sampling scheme. Under this scheme $f_t = 1$ and $g_t = \sqrt{2}$ in Eq. 1. Denote the backward time schedule as $\{t_j^{\leftarrow}\}_{0\leq j\leq N}$ such that $0 = t_0^{\leftarrow} < t_1^{\leftarrow} < \cdots, t_N^{\leftarrow} = T - \alpha$. To simulate the backward SDE, we use the exponential integrator scheme [31] which can be piecewisely expressed as a continuous-time SDE: for any $t \in [t_j^{\leftarrow}, t_{j+1}^{\leftarrow})$. .

$$d\bar{Y}_t = (\bar{Y}_t + 2s_{T-t_j^{\leftarrow},\theta_y}(\bar{Y}_{t_j^{\leftarrow}}))dt + \sqrt{2}d\bar{W}_t. \tag{15}$$

Denote $q_t(.|y) :=$ Law $(\bar{Y}_t), \forall t \in [0, T-\alpha]$. $\gamma_k = t_{k+1}^{\leftarrow} - t_k^{\leftarrow}$ and assume there exists $\kappa > 0$ such that $\gamma_k \leq \kappa \min\{1, T - t_{k+1}^{\leftarrow}\}$. Let u_2^2 be such that $\mathbb{E}_{x_0\sim P_0(.|y)}[\|x\|^2] \leq u_2^2 < \infty, \forall y \in \mathcal{Y}$.

Remark 1. *Assumption 1 ensure the data belong to a bounded set and the score is well defined. This also ensures the second moment of the data distribution are bound which is necessary for convergence of forward SDE. Some works [3, 29] do not require the existence of score function for the data distribution $P_0(.|y)$. These works employ early stopping of the reverse (sampling) process. They do so because for non-smooth data distributions $\nabla \log q_t$ can blow up as $t \to T$. This means that the model will approximate $q_{T-\alpha}$ rather than $q_T = P_0(.|y)$, which is acceptable since for small α the distance (e.g. in Wasserstein-p metric) between $q_{T-\alpha}$ and $P_0(.|y)$ is small [3].*

Having defined the generation algorithm and the assumptions required for our analysis, we present the main result of the paper.

Theorem 1. *Suppose Assumption 1 holds and $\delta > 0, \delta \ll 1$. Then with probability $1 - N(\sum_{y\in\mathcal{Y}} n_y)\delta$, we have*

1. *the empirical loss functions $\bar{\mathcal{L}}_{reg}^{n_y}(\theta_y, \theta_{-y})$, are $\frac{L_y}{m^2}$ smooth w.r.t to their own parameter $\theta_y \forall y \in \mathcal{Y}$ (See Lemma 3 in Appendix B.5)*

2. *If one runs individual gradient descent with step-size $\eta_\tau \leq \frac{m^2}{\max_{y\in\mathcal{Y}} L_y\sqrt{T_{train}}}$ for T_{train} iterations and selects the parameter from $(\theta_y^\tau, \theta_{-y}^\tau)_{\tau\in[T_{train}]}$ that minimizes the Nash Gap of the game $\langle \mathcal{Y}, (\bar{\mathcal{L}}_{reg}^{n_y})_{y\in\mathcal{Y}}, (\Theta_y)_{y\in\mathcal{Y}}\rangle$ and samples according to Eq.15, the sampling error*

$$\max_{y\in\mathcal{Y}} D_{KL}(P_\alpha(.|y)||q_{T-\alpha}(.|y)) \lesssim \max_{y\in\mathcal{Y}} \mathcal{L}^y(\bar{\theta}_y^*) + \tilde{\mathcal{O}}\left(\frac{m^2}{\sqrt{T_{train}}} + \beta\right) + \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{mn^*}} + \frac{1}{m}\right) + C_0(\kappa^2 N u_2^2 + \kappa T u_2^2 + \exp(-2T)u_2^2) + \bar{\mathcal{C}} \tag{16}$$

where $n^* = \min_{y\in\mathcal{Y}} n_y$, $\bar{\mathcal{C}} = \max_{y\in\mathcal{Y}} -\bar{C}(y)$ as in Eq. 11, $\kappa^2 N u_2^2 + \kappa T u_2^2$ is an upper bound on the discretization error due to the reverse SDE, $\exp(-2T)u_2^2$ is the error due to the convergence of the forward SDE and constant C_0 is some constant. $\tilde{\mathcal{O}}$ hides the $\log\frac{1}{\delta}$ factors, $|\mathcal{Y}|^2$ and bounds on strategy space, embedding matrices and other constants.

Putting all the details together, we get the following corollary.

Corollary 1 (Full Error Analysis). *Fix $\epsilon > 0$ arbitrarily. If $T \geq 1, \alpha < 1$ and $N > \log\frac{1}{\alpha}$, then there exists $0 = t_o < t_1 < \cdots t_N = T - \alpha$ such that for some $\kappa = \Theta(\frac{T+\log\frac{1}{\alpha}}{N})$ and $\gamma_k \leq \kappa \min\{1, T - t_k + 1\} \forall k = 0, 1, \cdots, N-1$. If we take $T = \frac{1}{2}\log\frac{d}{\epsilon}$, $N = \Theta(\frac{d(T+\log\frac{1}{\alpha})^2}{\epsilon})$, $\beta = \tilde{\Theta}(\epsilon)$, $T_{train} = \tilde{\Theta}(\frac{1}{\epsilon^6})$ and $m = \tilde{\Theta}(\frac{1}{\epsilon^2})$, then under similar conditions as Theorem 1, we achieve*

$$\max_{y\in\mathcal{Y}} D_{KL}(P_{\alpha(.|y)}||q_{T-\alpha}(.|y)) \lesssim \max_{y\in\mathcal{Y}} \mathcal{L}^y(\bar{\theta}_y^*) + \bar{\mathcal{C}} + \epsilon \tag{17}$$

where $\tilde{\mathcal{O}}, \tilde{\Theta}$ and $\lesssim$ hides the polynomial of $\log\frac{1}{\delta}$, $|\mathcal{Y}|^2$ and bounds on strategy space, embedding matrices and other constants. $\mathcal{L}^y(\bar{\theta}_y^*)$ is the universal approximation error of approximating the score with two layer network with random ReLUs.

Corollary 1 gives us the range of hyper-parameters such as width of hidden-layer, number of training steps, discretization of sampling, etc. to achieve worst case sampling error of $\mathcal{O}(\epsilon+1)$. The $\mathcal{O}(1)$ term $C(y)$ in Eq. 11, can be viewed as the error incurred due to diffusion model's nature in approximating $\nabla \log p_t(x_t|y)$ which is intractable by $\nabla \log p_t(x_t|x_0, y)$ with reverse SDE.

5.1 Proof sketch of Theorem 1

We provide a sketch for the proof and defer the details to the Appendix. We use a slight variant of [3, Theorem 1] (See Appendix B for more details) to upper bound the KL-divergence between the distribution approximated by our model and the ground truth to get

$$\max_{y\in\mathcal{Y}} D_{KL}(P_\alpha(.|y)||q_{T-\alpha}(.|y)) \le \max_{y\in\mathcal{Y}} \mathcal{L}^y(\theta_y) + C_0(\kappa^2 N u_2^2 + \kappa T u_2^2 + \exp(-2T)u_2^2). \quad (18)$$

We then perform the following decomposition for $\max_{y\in\mathcal{Y}} \mathcal{L}^y(\theta_y)$ (See Appendix B.1 for more details), where

$$\begin{aligned}\min_{\tau\in[T_{train}]} \max_{y\in\mathcal{Y}} \mathcal{L}^y(\theta_y^\tau) \le \max_{y\in Y} \mathcal{L}^y(\theta_y^*) + +2\max_{y\in Y} \sup_{(\theta_y,\theta_{-y})} \mid \mathcal{L}^y_{reg}(\theta_y,\theta_{-y}) - \bar{\mathcal{L}}^{n_y}_{reg}(\theta_y,\theta_{-y}) \mid \\ + \min_{\tau\in[T_{train}]} \text{NE-gap}(\theta_y^\tau,\theta_{-y}^\tau) + \beta \max_{y\in Y} \sup_{(\theta_y,\theta_{-y})} \mathcal{L}^y_{mut}(\theta_y,\theta_{-y}).\end{aligned} \quad (19)$$

Proposition 1 (Training and bounding the Nash Gap). *Suppose $\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y^\tau,\theta_{-y}^\tau)$ is $\frac{L_y}{m^2}$ smooth for all $y \in \mathcal{Y}$. Then by selecting a constant learning rate $\eta_\tau \le \frac{\eta}{\sqrt{T_{train}}} \le \frac{m^2}{\max_{y\in\mathcal{Y}} L_y\sqrt{T_{train}}}$ that depends on the total iteration T_{train}, we have*

$$\min_{\tau\in[T_{train}]} \textit{NE-gap}(\theta_y^\tau,\theta_{-y}^\tau) \lesssim \min_{\tau\in[T_{train}]} \max_{y\in\mathcal{Y}} \left\|\nabla_{\theta_y} \bar{\mathcal{L}}^{n_y}_{reg}(\theta_y^\tau,\theta_{-y}^\tau)\right\|^2 = \tilde{\mathcal{O}}(\frac{m^2}{\sqrt{T_{train}}} + \beta), \quad (20)$$

where $\tilde{\mathcal{O}}$ hides the $\log\frac{1}{\delta}$ factors.

The proof is presented in Appendix B.6. Proposition 1 gives a non-asymptotic first order convergence of the individual gradient descent. When no further assumption on the gradient mapping i.e. the vector of $(\nabla_{\theta_y}\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y^\tau,\theta_{-y}^\tau))_{y\in\mathcal{Y}}$ such as (strong) monotonicity of the game $\langle \mathcal{Y}, (\bar{\mathcal{L}}^{n_y}_{reg})_{y\in\mathcal{Y}}, (\Theta_y)_{y\in\mathcal{Y}}\rangle$ is considered, this is the best we can hope for. The iterate at which the minimum Nash Gap is achieved can be tracked by storing the parameters $(\theta_y^\tau,\theta_{-y}^\tau)$ for which the $\max_{y\in\mathcal{Y}} \left\|\nabla_{\theta_y}\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y^\tau,\theta_{-y}^\tau)\right\|^2$ is the least.

Monte-Carlo Estimate. To bound $\max_{y\in Y}\mathcal{L}^y(\theta_y^*)$, we employ ideas from [15, Lemma 6]. Informally (See Prop 2 in Appendix B.7), for $\delta > 0, \delta \ll 1$, with probability $1 - 2N|\mathcal{Y}|\delta$ we achieve

$$\max_{y\in Y} \mathcal{L}^y(\theta_y^*) \lesssim \max_{y\in\mathcal{Y}} \mathcal{L}^y(\bar{\theta}_y^*) + \tilde{\mathcal{O}}\left(\frac{1}{m}\right), \quad (21)$$

where $\tilde{\mathcal{O}}$ hides the $\log\frac{1}{\delta}$ factors. $\mathcal{L}^y(\bar{\theta}_y^*)$ is the error associated with approximating the score of the data using a two layer networks of random ReLUs.

Rademacher Complexity. Finally, we bound the generalization error (See Lemma 9 in Appendix B.8 for the derivation) by the Rademacher Complexity

$$\max_{y\in Y} \sup_{(\theta_y,\theta_{-y})} \mid \mathcal{L}^y_{reg}(\theta_y,\theta_{-y}) - \bar{\mathcal{L}}^{n_y}_{reg}(\theta_y,\theta_{-y}) \mid = \tilde{\mathcal{O}}\left(\frac{1}{\sqrt{mn^*}}\right) + \bar{\mathcal{C}} \quad (22)$$

where $n^* = \min_{y\in\mathcal{Y}} n_y$, $\bar{\mathcal{C}} = \max_{y\in\mathcal{Y}} -\bar{C}(y)$. $\tilde{\mathcal{O}}$ hides the $\log\frac{1}{\delta}$ factors, $|\mathcal{Y}|^2$ and bounds on strategy space, embedding matrices and other constants.

Bound on Mutual Learning Loss The final term in Eq. 19 $\max_{y\in Y}\sup_{(\theta_y,\theta_{-y})}\mathcal{L}^y_{mut}(\theta_y,\theta_{-y})$ is $\mathcal{O}(1)$ (See Lemma 6 in Appendix B.8).

5.2 Interpretation of the Main Result and Implications for Long-tailed Learning

Firstly, when the training objective function are nice, Proposition 1 shows that individual gradient descent employed in Deep Mutual Learning literature is seeking a Nash Equilibrium of an underlying game across different models. Second, when diffusion models are employed for long-tailed generation, Theorem 1 shows that a Nash equilibrium of an underlying game across conditional score network achieves an egalitarian solution w.r.t to sampling error. Our result give insight into the bottleneck process in diffusion generative modeling when faced with limited computing resources and long-tailed data. To the best of our knowledge, our result is the first to provide a comprehensive view of Deep Mutual Learning and long-tailed generation(learning) with diffusion models.

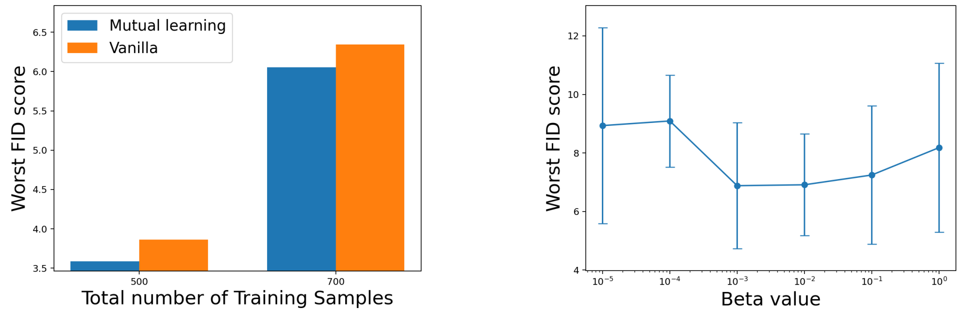

Figure 1: Left: Mutual Learning vs Vanilla Diffusion Mode ($\beta = 0.01, r = 1.2, T_{train} = 10000$). We plot the mean worst case FID score. Right: Graph of worst case FID score and varying β with mutual learning objective (No. of training samples= 100, $r = 2.5$.

6 Numerical Experiments

In this section, we numerically verify the theoretical results on synthetic datasets. We fix $d = 2$, the number of classes $|\mathcal{Y}| = 4$ and $r > 0$, such that the marginal density is set to $p(y) \propto y^{-r}, y \in \{1, 2, 3, 4\}$. $P(.|y), \forall y \in \mathcal{Y}$ is assumed to be $(\mathcal{N}(\mu_y, I_d))_{y \in \mathcal{Y}}$ whose mean points lie on a square of side length 2. The width of network $m = 16$, learning rate $\eta_\tau = 10^{-3}, \forall \tau, T_{train} = 10^4$ is fixed for Adam. We compute FID score between the true distribution and 100 generated samples (using probability flow ODE sampler from [15]) per class label (additional graphs are provided in Appendix C) for various cases. We repeat the generation of 100 samples 5 times with different initializations.

Mutual Learning vs Vanilla Diffusion Model. We fix $\beta = 10^{-3}$. Then, we compare our model with the vanilla case which is score based diffusion model without Deep Mutual Learning ($\beta = 0$) trained on each class label distribution and plotting the worst case FID score across class labels. We observe the mutual learning model outperforms the vanilla diffusion model in all the considered cases, as shown in Figure 1(Left).

Varying the regularizing parameter β. We fix the training dataset with size 100 and plot the worst case FID scores by varying β Figure 1(Right). We observe that the value of β should not be too low or too high in line with [21, Figure 5].

7 Discussion

Choice of $\lambda(t)$ and $\omega(t)$. We choose $\omega(t)$ as an increasing function of t (as in [21]) and $\lambda(t)$ such that $\frac{\lambda(t)}{\bar{\sigma}_t}$ is non-increasing in t. The motivation behind this is to ensure that the training process gives more weight to fitting to the data distribution for smaller $0 < t < T$ and give more weight to the mutual learning objective for high noise regions i.e. larger $0 < t < T$ of the forward diffusion process. There might exist a better weighting function. Our analysis doesn't involve the investigation of an optimal weighting function. We leave this as a future direction to pursue.

Bound on Approximation Error $\mathcal{L}^y(\bar{\theta}_y^*)$. We are aware of universal approximation results for two-Layer networks of Random ReLUs such as [9, Theorem 3.6]. Assuming $\nabla \log p_t(x_t|y)$ being Lipschitz continuous w.r.t. to x_t, we can also achieve a upper bound for $\max_{y \in \mathcal{Y}} \mathcal{L}^y(\bar{\theta}_y^*)$ as in [9] that can be made arbitrarily small by controlling the hyperparameters (such as bound on RKHS norm of $\bar{s}_{t,\bar{\theta}_y}$ and $0 < \delta \ll 1$).

Limitations. While we achieve a bound on the worst case generalization (sampling) error, it would be essential to extend the current analysis and provide insight into whether the performance of the head class score networks are preserved upon adding the mutual learning loss. Further, we set $\mathcal{Q} = Uniform(\mathcal{Y})$ as in mutual learning literature. Our analysis does not not reveal the effect of the distribution $\mathcal{Q}$ on the worst-case sampling error, urging further investigation. Additionally, it is worth examining if generalization (sampling) error can be made arbitrarily small (also noted in [29, Section 3.3]) i.e. the $\mathcal{O}(1)$ bias be removed. Finally, while given the theoretical nature of this work, we support our theoretical analysis with Gaussian data distribution, validating our findings on real world datasets such as CIFAR100LT/CIFAR10LT with complex neural network architecture could further strengthen the approach of Deep Mutual Learning for long-tailed generative modeling.

References

[1] Amira Alakhdar, Barnabas Poczos, and Newell Washburn. Diffusion models in de novo drug design. *Journal of Chemical Information and Modeling*, 64(19):7238–7256, 2024.

[2] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*, 2021.

[3] Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Nearly d-linear convergence bounds for diffusion models via stochastic localization. *arXiv preprint arXiv:2308.03686*, 2023.

[4] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a Nash equilibrium. *Communications of the ACM*, 52(2):89–97, 2009.

[5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[6] Hengyu Fu, Zhuoran Yang, Mengdi Wang, and Minshuo Chen. Unveil conditional diffusion models with classifier-free guidance: A sharp statistical theory. *arXiv preprint arXiv:2403.11968*, 2024.

[7] Zhiye Guo, Jian Liu, Yanli Wang, Mengrui Chen, Duolin Wang, Dong Xu, and Jianlin Cheng. Diffusion models in bioinformatics and computational biology. *Nature reviews bioengineering*, 2(2):136–154, 2024.

[8] Pengxiao Han, Changkun Ye, Jieming Zhou, Jing Zhang, Jie Hong, and Xuesong Li. Latent-based diffusion model for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2639–2648, 2024.

[9] Yinbin Han, Meisam Razaviyayn, and Renyuan Xu. Neural network-based score estimation in diffusion models: Optimization and generalization. *arXiv preprint arXiv:2401.15604*, 2024.

[10] Daniel Hsu, Clayton H Sanford, Rocco Servedio, and Emmanouil Vasileios Vlatakis-Gkaragkounis. On the approximation power of two-layer networks of random relus. In *Conference on Learning Theory*, pages 2423–2461. PMLR, 2021.

[11] Weipeng Fuzzy Huang, Junjie Tao, Changbo Deng, Ming Fan, Wenqiang Wan, Qi Xiong, and Guangyuan Piao. Rényi divergence deep mutual learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 156–172. Springer, 2023.

[12] Ziyu Jiang, Tianlong Chen, Bobak J Mortazavi, and Zhangyang Wang. Self-damaging contrastive learning. In *International Conference on Machine Learning*, pages 4927–4939. PMLR, 2021.

[13] Amirhossein Kazerouni, Ehsan Khodapanah Aghdam, Moein Heidari, Reza Azad, Mohsen Fayyaz, Ilker Hacihaliloglu, and Dorit Merhof. Diffusion models in medical imaging: A comprehensive survey. *Medical image analysis*, 88:102846, 2023.

[14] Jianing Li, Chaoqun Yang, Guanhua Ye, and Quoc Viet Hung Nguyen. Graph neural networks with deep mutual learning for designing multi-modal recommendation systems. *Information Sciences*, 654:119815, 2024.

[15] Puheng Li, Zhong Li, Huishuai Zhang, and Jiang Bian. On the generalization properties of diffusion models. *Advances in Neural Information Processing Systems*, 36:2097–2127, 2023.

[16] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Open long-tailed recognition in a dynamic world. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(3):1836–1851, 2022.

[17] Andreas Maurer. A vector-contraction inequality for rademacher complexities. In *Algorithmic Learning Theory: 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings 27*, pages 3–17. Springer, 2016.

[18] Christos H Papadimitriou, Emmanouil-Vasileios Vlatakis-Gkaragkounis, and Manolis Zampetakis. The computational complexity of multi-player concave games and kakutani fixed points. *arXiv preprint arXiv:2207.07557*, 2022.

[19] Changhwa Park, Junho Yim, and Eunji Jun. Mutual learning for long-tailed recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2675–2684, 2023.

[20] Wonpyo Park, Wonjae Kim, Kihyun You, and Minsu Cho. Diversified mutual learning for deep metric learning. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 709–725. Springer, 2020.

[21] Yiming Qin, Huangjie Zheng, Jiangchao Yao, Mingyuan Zhou, and Ya Zhang. Class-balancing diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18434–18443, 2023.

[22] Yiming Qin, Huangjie Zheng, Jiangchao Yao, Mingyuan Zhou, and Ya Zhang. Class-Balancing Diffusion Models (arxiv version), 2023.

[23] Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. *Advances in neural information processing systems*, 21, 2008.

[24] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International conference on machine learning*, pages 8857–8868. PMLR, 2021.

[25] Jie Shao, Ke Zhu, Hanxiao Zhang, and Jianxin Wu. Diffult: Diffusion for long-tail recognition without external knowledge. *Advances in Neural Information Processing Systems*, 37:123007–123031, 2024.

[26] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021.

[27] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[28] Shuhan Tan, Yujun Shen, and Bolei Zhou. Improving the fairness of deep generative models without retraining. *arXiv preprint arXiv:2012.04842*, 2020.

[29] Yuqing Wang, Ye He, and Molei Tao. Evaluating the design space of diffusion-based generative models. *arXiv preprint arXiv:2406.12839*, 2024.

[30] Divin Yan, Lu Qi, Vincent Tao Hu, Ming-Hsuan Yang, and Meng Tang. Training class-imbalanced diffusion model via overlap optimization. *arXiv preprint arXiv:2402.10821*, 2024.

[31] Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022.

[32] Tianjiao Zhang, Huangjie Zheng, Jiangchao Yao, Xiangfeng Wang, Mingyuan Zhou, Ya Zhang, and Yanfeng Wang. Long-tailed diffusion models with oriented calibration. In *The twelfth international conference on learning representations*, 2024.

[33] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4320–4328, 2018.

[34] Haojie Zhao, Gang Yang, Dong Wang, and Huchuan Lu. Deep mutual learning for visual object tracking. *Pattern Recognition*, 112:107796, 2021.

[35] Yuanzhi Zhu, Zhaohai Li, Tianwei Wang, Mengchao He, and Cong Yao. Conditional text image generation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14235–14245, 2023.

[36] Hao Zou, Zae Myung Kim, and Dongyeop Kang. A survey of diffusion models in natural language processing. *arXiv preprint arXiv:2305.14671*, 2023.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: **We discuss under related works, the scope and how our work differs from those in existing literature. Theorem 1, Proposition 1, Corollary 1 along with numerical experiments reflect the theoretical contribution of our paper and support the claims made in the abstract and introduction .**

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: **In discussion section, we highlight limitations of our theoretical analysis.**

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: **We clearly state the assumptions we make and cite references that use existing literature. We provide a proof sketch of the Main Theorem of the paper. The complete proof and supporting lemmas have been cited are provided in the supplemantary material.**

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: **We provide the hyper-parameter values and provide additional graphs in Appendix C**.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: **For the purpose of blind review, to preserve anonymity we provide the code only with the supplementary material for the reviewers to access. We assure the reviewers that we will provide open access to code once the review process is complete.**

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: **We clearly state the dataset chosen and the hyper-parameters used for the experiments.**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: **We provide error bar of 1 standard deviation and provide the experimental setting in the main text.**

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: **We provide the details of our computing resources in the supplementary material in Appendix C**

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: **We confirm that our research is in line with NeurIPS Code of Ethics. We perform our experiments on synthetic datasets.**

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: **We point out in our abstract, introduction and discussion the broader impacts of our work to privacy, copyright related issues in generative modeling and existing literature on diffusion models for Long-tailed generation.**

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: **We do not release any model along with our work. Our research is focussed on the theoretical analysis of diffusion models for long-tailed generation and our numerical experiments are conducted on sythetic datasets to validate our theoretical findings and not on real world data.**

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: **We only use the probability flow ODE sampler used in [15] and we mention this in the main text under numerical experiments.**

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: **We use simple synthetic data to verify our theoretical findings. We do not release any new asset. We provide our code with supplementary materials. We shall create a github repository for the code if the submission is accepted.**

Guidelines:

- The answer NA means that the paper does not release new assets.

- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: **No crowd sourcing experiments or research with human subjects were conducted for this work.**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: **The numerical experiments were performed on synthetic data. There were no subjects on which experiments were conducted.**

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: **LLM wasn't use for any task during the development of the research or the preparation of the manuscript.**

    Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Appendix / supplemental material

Notations	
Symbol	Meaning
m	Number of neurons in hidden-layer of score network
C_{w_y,u_y}	Upper bound on $\|w_{y,i}\|_1, \|u_{y,i}\|_1$
F_T^2	Upper bound on $\mathbb{E}_{X_0}\mathbb{E}_{\xi_j}\left[\|\sigma(Wx(t)+Ue(t))\|_2^2\right], 0 \leq t \leq T$
L_y	$\bar{\mathcal{L}}_{reg}^{n_y}(\theta_y, \theta_{-y})$ is $\frac{L_y}{m^2}$ smooth w.r.t. θ_y
ϕ_y	Lipschitz constant of $\bar{\mathcal{L}}_{mut}^{n_y}(\theta_y, \theta_{-y})$ w.r.t. θ_y
σ_y	$\bar{\mathcal{L}}_{reg}^{n_y}(\theta_y, \theta_{-y})$ is σ_y Lipschitz in θ_y
B	Upper Bound of the Frobenius norm of A_y

B Sampling

Denote the backward time schedule as $\{t_j^{\leftarrow}\}_{0\leq j\leq N}$ such that $0 = t_0^{\leftarrow} < t_1^{\leftarrow} < \cdots, t_N^{\leftarrow} = T-\alpha$. Lower case p_t represents the density of P_t. We consider the exponential integrator scheme for simulating the backward SDE with

The generation algorithm can be expressed as a piecewise continuous-time SDE: for any $t \in [t_j^{\leftarrow}, t_{j+1}^{\leftarrow})$.

$$d\bar{Y}_t = (\bar{Y}_t + 2s_{T-t_j^{\leftarrow},\theta_y}(\bar{Y}_{t_j^{\leftarrow}}))dt + \sqrt{2}d\bar{W}_t \tag{23}$$

Denote $q_t :=$ Law $(\bar{Y}_t), \forall t \in [0, T-\delta]$.

Theorem 2. *[3, Theorem 1] Let Assumption 1 hold. Then there exists a numerical constant $C_0 > 0$, such that*

$$D_{KL}(p_\alpha(.|y)||q_{T-\alpha}(.|y)) \leq C_0(E_S + E_D + E_F) \tag{24}$$

where $E_D \leq \kappa^2 N u_2^2 + \kappa T u_2^2$ is the discretization error due to the reverse SDE, $E_F \leq \exp(-2T)u_2^2$ is the error due to the convergence of the forward SDE and E_S is the score estimation error

$$E_S(\theta_y) = \sum_{j=0}^{N-1} \gamma_j \mathbb{E}_{x\sim p_{T-t_j^{\leftarrow}}}\left[\left\|\nabla \log p_{T-t_j^{\leftarrow}}(x|y) - s_{T-t_j^{\leftarrow},\theta_y}(x)\right\|_2^2\right] \tag{25}$$

where $\gamma_j := t_{j+1}^{\leftarrow} - t_j^{\leftarrow}, \forall j = 0, 1, \cdots N-1$ is the step-size of the generation algorithm.

When the training is done over the forward discretization given by $(t_{N-j} = T - t_j^{\leftarrow})_{j=0}^{N-1}$, we have

$$\begin{aligned} E_S = &\sum_{j=0}^{N-1} \frac{\bar{\sigma}_{t_{N-j}}\lambda(t_{N-j})}{\lambda(t_{N-j})\bar{\sigma}_{t_{N-j}}}(t_{N-j} - t_{N-j-1})\mathbb{E}_{X_0}\mathbb{E}_{X_{t_{N-j}}|X_0}\left\|\bar{\sigma}_{t_{N-j}}s_{t_{N-j},\theta_y}(X_{t_{N-j}}) + \xi\right\|^2 \\ &+ \sum_{j=0}^{N-1}\frac{\bar{\sigma}_{t_{N-j}}}{\lambda(t_{N-j})}\lambda(t_{N-j})(t_{N-j} - t_{N-j-1})C_{t_{N-j}} \\ \leq &2\max_j \frac{\bar{\sigma}_{t_{N-j}}}{\lambda(t_{N-j})}\mathcal{L}^y(\theta_y) \end{aligned}$$

where

$$\begin{aligned} \mathcal{L}^y(\theta_y) = \frac{1}{2}\sum_{j=0}^{N-1}\mathbb{E}_{X_0}\mathbb{E}_{X_{t_{N-j}}|X_0}\Big[&\lambda(t_{N-j})(t_{N-j} - t_{N-j-1}) \\ &\left\|\nabla_{x(t_{N-j})}\log p_{t_{N-j}}(x(t_{N-j})|x_0) - s_{t_{N-j},\theta_y}(x(t_{N-j}))\right\|_2^2\Big] \\ &+ \frac{1}{2}\sum_{j=0}^{N-1}\lambda(t_{N-j})(t_{N-j} - t_{N-j-1})C_{t_{N-j}}(y) \end{aligned} \tag{26}$$

Theorem 3. *(Appendix B and [3, Theorem 1]) Let Assumption 1 hold. Then there exists a numerical constant $C_0 > 0$, such that*

$$D_{KL}(p_\alpha(.|y)||q_{T-\alpha}(.|y)) \leq C_0(\mathcal{L}^y(\theta_y) + E_D + E_F) \tag{27}$$

where $E_D \leq \kappa^2 N u_2^2 + \kappa T u_2^2$ is the discretization error due to the reverse SDE, $E_F \leq \exp(-2T)u_2^2$ is the error due to the convergence of the forward SDE.

B.1 Decomposition of $\mathcal{L}^y(\theta_y)$

Let $\theta_y^* = \operatorname{argmin}_{\theta_y} \mathcal{L}^y(\theta_y)$. We further decompose $\mathcal{L}^y(\theta_y)$ as

$$\begin{aligned}
\max_{y\in Y}\left(\mathcal{L}^y(\theta_y) - \mathcal{L}^y(\theta_y^*)\right) &\leq \max_{y\in Y}\Big(\mathcal{L}^y(\theta_y) + \beta\mathcal{L}^y_{mut}(\theta_y,\theta_{-y}) - (\mathcal{L}^y(\theta_y^*) + \beta\mathcal{L}^y_{mut}(\theta_y^*,\theta_{-y}))) \\
&\quad +\beta(\mathcal{L}^y_{mut}(\theta_y^*,\theta_{-y}) - \mathcal{L}^y_{mut}(\theta_y,\theta_{-y}))\Big) \\
&\overset{(a)}{\leq} \max_{y\in Y}\Big(\mathcal{L}^y(\theta_y) + \beta\mathcal{L}^y_{mut}(\theta_y,\theta_{-y}) - (\mathcal{L}^y(B(\theta_{-y})) \\
&\quad + \beta\mathcal{L}^y_{mut}(B(\theta_{-y}),\theta_{-y}))\Big) + \beta\max_{y\in Y}\mathcal{L}^y_{mut}(B(\theta_{-y}),\theta_{-y}) \\
&\leq \max_{y\in Y}\left(\mathcal{L}^y_{reg}(\theta_y,\theta_{-y}) - \mathcal{L}^y_{reg}(B(\theta_{-y}),\theta_{-y})\right) \\
&\quad + \beta\max_{y\in Y}\sup_{(\theta_y,\theta_{-y})}\mathcal{L}^y_{mut}(\theta_y,\theta_{-y})
\end{aligned}$$

where (a) follows from the fact that $\mathcal{L}^y_{reg}(B(\theta_{-y}),\theta_{-y}) \leq \mathcal{L}^y_{reg}(\theta_y,\theta_{-y})$. We further decompose this to obtain an upper bound on $\max_{y\in Y}\min_t \mathcal{L}^y(\theta_y^t)$

$$\begin{aligned}
\max_{y\in Y}\mathcal{L}^y(\theta_y) &\leq \max_{y\in Y}\mathcal{L}^y(\theta_y^*) + \max_{y\in Y}\left(\mathcal{L}^y_{reg}(\theta_y,\theta_{-y}) - \mathcal{L}^y_{reg}(B(\theta_{-y}),\theta_{-y})\right) \\
&\quad + \beta\max_{y\in Y}\sup_{(\theta_y,\theta_{-y})}\mathcal{L}^y_{mut}(\theta_y,\theta_{-y}) \\
&\overset{(a)}{\leq} \max_{y\in Y}\mathcal{L}^y(\theta_y^*) + \max_{y\in Y}\mid \mathcal{L}^y_{reg}(\theta_y,\theta_{-y}) - \bar{\mathcal{L}}^{n_y}_{reg}(\theta_y,\theta_{-y})\mid \\
&\quad + \max_{y\in Y}\mid \bar{\mathcal{L}}^{n_y}_{reg}(\theta_y,\theta_{-y}) - \bar{\mathcal{L}}^{n_y}_{reg}(B(\theta_{-y}),\theta_{-y})\mid \\
&\quad + \max_{y\in Y}\mid \mathcal{L}^y_{reg}(B(\theta_{-y}),\theta_{-y}) - \bar{\mathcal{L}}^{n_y}_{reg}(B(\theta_{-y}),\theta_{-y})\mid \\
&\quad + \beta\max_{y\in Y}\sup_{(\theta_y,\theta_{-y})}\mathcal{L}^y_{mut}(\theta_y,\theta_{-y}) \\
&\overset{(b)}{\leq} \max_{y\in Y}\mathcal{L}^y(\theta_y^*) + 2\max_{y\in Y}\sup_{(\theta_y,\theta_{-y})}\mid \mathcal{L}^y_{reg}(\theta_y,\theta_{-y}) - \bar{\mathcal{L}}^{n_y}_{reg}(\theta_y,\theta_{-y})\mid \\
&\quad + \text{NE-gap}(\theta_y,\theta_{-y}) + \beta\max_{y\in Y}\sup_{(\theta_y,\theta_{-y})}\mathcal{L}^y_{mut}(\theta_y,\theta_{-y}) \\
\implies \min_{\tau\in[T_{train}]}\max_{y\in\mathcal{Y}}\mathcal{L}^y(\theta_y^\tau) &\overset{(c)}{\leq} \max_{y\in Y}\mathcal{L}^y(\theta_y^*) + \\
&\quad + 2\max_{y\in Y}\sup_{(\theta_y,\theta_{-y})}\mid \mathcal{L}^y_{reg}(\theta_y,\theta_{-y}) - \bar{\mathcal{L}}^{n_y}_{reg}(\theta_y,\theta_{-y})\mid \\
&\quad + \min_{\tau\in[T_{train}]}\text{NE-gap}(\theta_y^\tau,\theta_{-y}^\tau) + \beta\max_{y\in Y}\sup_{(\theta_y,\theta_{-y})}\mathcal{L}^y_{mut}(\theta_y,\theta_{-y})
\end{aligned}$$

where (a) follows from adding and subtracting the empirical losses $\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y,\theta_{-y})$ and $\bar{\mathcal{L}}^{n_y}_{reg}(B_y(\theta_{-y}),\theta_{-y})$ and using triangle inequality of the max norm, (b) follows from the gradient domination property for strongly convex functions, (c) follows from taking the minimum over the iterates of the algorithm.

B.2 Boundedness of Forward Dynamics

Lemma 1. *Consider the forward diffusion process with linear drift coefficients. For any $\delta > 0, \delta \ll 1$, w.p. (with probability) of atleast $1-\delta$. we have*

$$\|x(t)\|_\infty \leq C_T\Bigg(\|x(0)\|_\infty + \sqrt{\log \frac{2}{\pi\delta^2}}\Bigg) \tag{28}$$

where $C_T := \max_{t\in[0,T]} r(t), r(t)v(t)$.

Proof: The proof is similar to [15, Lemma 1] When the drift coefficient $f(.,t) : \mathbb{R}^d \to \mathbb{R}^d$ is linear in x i.e. $f(x,t) = -f(t)x$, the transition kernel $p_{t|0}$ has a closed form

$$p_{t|0}(x(t)|x(0)) = \mathcal{N}(x(t); \mu(t)x(0), \bar{\sigma}^2(t)I_d) \tag{29}$$

where $\mu(t) := \exp\left(\int_0^t f(\xi)d\xi\right), \bar{\sigma}^2(t) := 2\int_0^t \exp\left(2\mu_s - 2\mu_t\right)\sigma_s^2 ds$. Together we get,

$$x(t) = \mu(t)x(0) + \bar{\sigma}(t)z, z \sim \mathcal{N}(0, I_d) \tag{30}$$

For any $\epsilon \sim \mathcal{N}(0,1), c > 1$, we have

$$\mathbb{P}\{\epsilon : |\epsilon| > c\} = 2\int_c^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}dx \leq \frac{1}{\sqrt{2\pi}}\int_c^\infty 2xe^{-\frac{x^2}{2}}dx = \frac{1}{\sqrt{2\pi}}\int_{c^2}^\infty e^{-\frac{x}{2}}dx = \sqrt{\frac{2}{\pi}}e^{-\frac{c^2}{2}} \tag{31}$$

Let $\delta = \sqrt{\frac{2}{\pi}}e^{-\frac{c^2}{2}}$, then

$$\mathbb{P}\{\epsilon : |\epsilon| \leq \sqrt{\log\frac{2}{\pi\delta^2}}\} \geq 1-\delta \tag{32}$$

Hence, for any $\delta \in (0,1)$ with $\delta \ll 1$, w.p. at least $1-\delta$, we have

$$\|x(t)\|_\infty \leq C_T\Bigg(\|x(0)\|_\infty + \sqrt{\log \frac{2}{\pi\delta^2}}\Bigg) \tag{33}$$

where $C_T := \max_{t\in[0,T]}\{\mu(t), \bar{\sigma}(t)\}$. Let $C_{T,\delta} = C_T(K + \sqrt{\log\frac{2}{\pi\delta^2}})$

B.3 Boundedness of Loss function $\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y})$

In this section, study some properties of the game defined by $\langle \mathcal{Y}, (\bar{\mathcal{L}}^{n_y}_{reg})_{y\in\mathcal{Y}}, (\Theta_y)_{y\in\mathcal{Y}}\rangle$. From Eq. 10, we have

$$\mathcal{L}^y_{conti,reg}(\theta_y, \theta_{-y}) = \mathcal{L}^y_{conti}(\theta_y) + \beta\mathcal{L}^y_{conti,mut}(\theta_y, \theta_{-y}) \tag{34}$$

where

$$\mathcal{L}^y_{conti,mut}(\theta_y, \theta_{-y}, \omega(.)) = \frac{1}{2}\int_{t_0}^T \omega(t)\mathbb{E}_{x(t)\sim p_t}\mathbb{E}_{y'\sim Q}\left[\left\|s_{t,\theta_y}(x(t)) - s_{t,\theta_{y'}}(x(t))\right\|_2^2\right]dt$$

and

$$\mathcal{L}^y_{conti}(\theta_y, \theta_{-y}) = \frac{1}{2}\int_{t_0}^T \lambda(t)\mathbb{E}_{(x(t),y)}\left[\left\|\nabla_{x(t)}\log p_t(x(t)|y) - s_{t,\theta}(x(t),y)\right\|_2^2\right]dt$$

Conditioning on X_0 and using law of iterated expectation, we can write [29, Appendix A], we get

$$\begin{aligned}\mathcal{L}^y_{conti,reg}(\theta_y, \theta_{-y}) =& \frac{1}{2}\int_{t_0}^T \mathbb{E}_{X_0}\mathbb{E}_{X_t|X_0,y}\Big[\lambda(t)\left\|s_{t,\theta_y}(x(t)) - \nabla_{x(t)}\log p_t(x(t)|x_0)\right\|_2^2 \\ &+ \beta\omega(t)\mathbb{E}_{y'\sim Q}\left[\left\|s_{t,\theta_y}(x(t)) - s_{t,\theta_{y'}}(x(t))\right\|_2^2\right]\Big]dt + \frac{1}{2}\int_{t_0}^T\Big[\lambda(t)C_t(y)\Big]dt\end{aligned}$$

where $C_t(y) = \mathbb{E}_{X_t} \left\| \nabla \log p_t(X_t|y) \right\|^2 - \mathbb{E}_{X_0}\mathbb{E}_{X_t|X_0} \left\| \nabla \log p_t(X_t|X_0, y) \right\|^2$ to learn the score $\nabla_{x(t)} \log p_t(x(t)|x_0, y)$.

Furthermore, we discretize the time points $0 = t_0 < t_1 < \cdots < t_N = T$ to the objective function

$$\begin{aligned}
\mathcal{L}^y_{reg}(\theta_y, \theta_{-y}) &= \mathcal{L}^y(\theta_y) + \beta \mathcal{L}^y_{mut}(\theta_y, \theta_{-y}) \\
&= \frac{1}{2} \sum_{j=1}^{N} \lambda(t_j)(t_j - t_{j-1}) \mathbb{E}_{X_0} \mathbb{E}_{X_{t_j}|X_0} \left[\left\| \nabla_{x(t_j)} \log p_t(x_i(t_j)|x_0) - s_{t_j,\theta_y}(x_i(t_j)) \right\|_2^2 \right] + \\
&+ \bar{C}(y) + \beta \frac{1}{2} \sum_{j=1}^{N} \omega(t_j)(t_j - t_{j-1}) \mathbb{E}_{X_0} \mathbb{E}_{X_{t_j}|X_0} \mathbb{E}_{y' \sim Q} \left[\left\| s_{t_j,\theta_y}(x_i(t_j)) - s_{t,\theta_{y'}}(x_i(t_j)) \right\|_2^2 \right]
\end{aligned} \tag{35}$$

where $\bar{C}(y) = \frac{1}{2} \sum_{j=1}^{N} \lambda(t_j)(t_j - t_{j-1}) C_{t_j}(y)$ From [29, Appendix A], we have $X_t|X_0 \sim \mathcal{N}(e^{-\mu_t} X_0, \bar{\sigma}_t^2 I)$ and its density function is

$$p_t(x|x_0) = (2\pi\bar{\sigma}_t^2)^{-\frac{d}{2}} \exp(-\frac{\|x - e^{-\mu_t} x_0\|}{2\bar{\sigma}_t^2})$$

Then,

$$\begin{aligned}
\Delta =& \mathbb{E}_{X_0} \mathbb{E}_{X_t|X_0} \left\| s_{t_j,\theta_y}(x_i(t_j)) - \nabla_{x(t_j)} \log p_t(x(t_j)|x_0) \right\| \\
=& \mathbb{E}_{X_0} \mathbb{E}_{X_t|X_0} \left\| s_{t_j,\theta_y}(x_i(t_j)) - \nabla_x \left(- \frac{\|X_t - e^{-\mu_t} X_0\|}{2\bar{\sigma}_t^2} \right) \right\|^2 \\
=& \mathbb{E}_{X_0} \mathbb{E}_{X_t|X_0} \left\| s_{t_j,\theta_y}(x_i(t_j)) + \frac{X_t - e^{\mu_t} X_0}{\bar{\sigma}_t^2} \right\|^2 \\
=& \mathbb{E}_{X_0} \mathbb{E}_{\epsilon_t} \left\| s_{t_j,\theta_y}(x_i(t_j)) + \frac{\epsilon_t}{\bar{\sigma}_t^2} \right\|^2
\end{aligned}$$

Let $\xi = \frac{\epsilon_t}{\bar{\sigma}_t} \sim \mathcal{N}(0, I)$

$$\Delta = \frac{1}{\bar{\sigma}_t} \mathbb{E}_{X_0} \mathbb{E}_{\xi} \left\| \bar{\sigma}_t s_{t_j,\theta_y}(x_i(t_j)) + \xi \right\|^2 \tag{36}$$

Finally putting all of it together, we get the empirical loss function

$$\begin{aligned}
\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y}) =& \frac{1}{2n_y} \sum_{i=1}^{n_y} \sum_{j=1}^{N} \frac{\lambda(t_j)(t_j - t_{j-1})}{\bar{\sigma}_{t_j}} \left[\left\| \bar{\sigma}_{t_j} s_{t_j,\theta_y}(x_i(t_j)) + \xi_{ij} \right\|_2^2 \right. \\
& \left. + \beta \omega(t_j)(t_j - t_{j-1}) \mathbb{E}_{y' \sim Q} \left[\left\| s_{t_j,\theta_y}(x_i(t_j)) - s_{t,\theta_{y'}}(x_i(t_j)) \right\|_2^2 \right] \right]
\end{aligned} \tag{37}$$

We will show that the empirical loss function for the label $y \in \mathcal{Y}$, $\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y})$ that is optimized is convex and smooth in θ_y with high probability.

Lemma 2. *For $\delta > 0, \delta \ll 1, wp. 1 - n_y N\delta$, the empirical loss function*

$$\begin{aligned}
\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y}) =& \frac{1}{2n_y} \sum_{i=1}^{n_y} \sum_{j=1}^{N} \frac{\lambda(t_j)(t_j - t_{j-1})}{\bar{\sigma}_{t_j}} \left[\left\| \bar{\sigma}_{t_j} s_{t_j,\theta_y}(x_i(t_j)) + \xi_{ij} \right\|_2^2 \right. \\
& \left. + \beta \omega(t_j)(t_j - t_{j-1}) \mathbb{E}_{y' \sim Q} \left[\left\| s_{t_j,\theta_y}(x_i(t_j)) - s_{t,\theta_{y'}}(x_i(t_j)) \right\|_2^2 \right] \right]
\end{aligned} \tag{38}$$

is bounded i.e.

$$\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y}) = \mathcal{O}(\sum_{j=1}^{N} \frac{\lambda(t_j)(t_j - t_{j-1})}{\bar{\sigma}_{t_j}} + \beta \omega(t_j)(t_j - t_{j-1}))$$

Proof: From Lemma 1, we have $\delta > 0, \delta \ll 1$

$$\mathbb{P}\{|\xi_{ij}| > \sqrt{\frac{2}{\pi\delta^2}}\} \leq \delta \tag{39}$$

Thus, $w.p.\ 1-n_yN\delta$, we have $|\xi_{ij}| \leq \sqrt{\frac{2}{\pi\delta^2}}$ and hence we have $\|x(t_j)\|_\infty \leq C_{t_N,\delta}, \forall i = 1, \cdots, n_y$ and $j = 1, , \cdots, N$

Thus, $w.p.\ 1 - n_yN\delta$

$$\begin{aligned}
\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y}) =& \frac{1}{2n_y}\sum_{i=1}^{n_y}\sum_{j=1}^{N}\frac{\lambda(t_j)(t_j - t_{j-1})}{\bar{\sigma}_{t_j}}\Big[\left\|\bar{\sigma}_{t_j}s_{t_j,\theta_y}(x_i(t_j)) + \xi_{ij}\right\|_2^2 \\
&+ \beta\omega(t_j)(t_j - t_{j-1})\mathbb{E}_{y'\sim Q}\Big[\left\|s_{t_j,\theta_y}(x_i(t_j)) - s_{t,\theta_{y'}}(x_i(t_j))\right\|_2^2\Big]\Big] \\
\leq& \frac{1}{n_y}\sum_{i=1}^{n_y}\sum_{j=1}^{N}\frac{\lambda(t_j)(t_j - t_{j-1})}{\bar{\sigma}_{t_j}}(\bar{\sigma}^2_{t_j}\left\|s_{t_j,\theta_y}(x_i(t_j))\right\|_2^2 + \|\xi_{ij}\|^2) \\
&+ \beta\omega(t_j)(t_j - t_{j-1})(\left\|s_{t_j,\theta_y}(x_i(t_j))\right\|_2^2 + \max_{y'\in\mathcal{Y}}\left\|s_{t_j,\theta_{y'}}(x_i(t_j))\right\|_2^2)
\end{aligned}$$

For a bound on $\left\|s_{t_j,\theta_y}(x(t_j))\right\|_2$

$$\begin{aligned}
\left\|s_{t_j,\theta_y}(x(t_j))\right\|_2 &= \left\|\frac{1}{m}\sum_{i=1}^{m}a_{y,i}\sigma(w_{y,i}^Tx(t_j) + u_{y,i}^Te(t_j))\right\|_2 && (40)\\
&\overset{(a)}{\leq} \frac{1}{m}\sum_{i=1}^{m}\|a_{y,i}\|_2\,|\sigma(w_{y,i}^Tx(t_j) + u_{y,i}^Te(t))| && (41)\\
&\overset{(b)}{\leq} \frac{1}{m}\sum_{i=1}^{m}\|a_{y,i}\|_2\,(\|w_{y,i}\|_1\,\|x(t_j)\|_\infty + \|u_{y,i}\|_1\,\|e(t))\|_\infty) && (42)\\
&\leq \frac{1}{m}\sum_{i=1}^{m}\|a_{y,i}\|_2\,(C_{t_N,\delta}\,\|w_{y,i}\|_1 + \max_j\|e(t_j)\|_\infty\,\|u_{y,i}\|_1) && (43)\\
&\overset{(c)}{\leq} (C_{t_n,\delta} + C_{t_N,e})C_{w_y,u_y}B && (44)
\end{aligned}$$

where (a) follows from triangle inequality for norms, (b) follows from the fact that the ReLu function satisfies $|\sigma(x)| \leq |x|$ and Holder inequality and (c) follows from the bounds on the embeddings and $x(t_j)$ with $\|w_{y,i}\|_1, \|u_{y,i}\|_1 \leq C_{w_y,u_y}, \forall i \in [m]$. Thus, for $\delta > 0, \delta \ll 1$, we have $w.p.\ 1 - n_yN\delta$

$$\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y}) \leq C_1\sum_{j=1}^{N}\frac{\lambda(t_j)(t_j - t_{j-1})}{\bar{\sigma}_{t_j}} + \beta\omega(t_j)(t_j - t_{j-1}) \tag{45}$$

where $C_1 = (\bar{\sigma}^2_{t_N} + 2)(C_{t_n,\delta} + C_{t_N,e})^2C^2_{w_y,u_y}B^2 + \frac{2}{\pi\delta^2}$. Since $\bar{\sigma}_{t_j}$ is non-decreasing in j, so $\max_j\bar{\sigma}_{t_j} = \bar{\sigma}_{t_N}$.

B.4 Boundedness of Gradient of Loss function $\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y})$

$$
\begin{aligned}
&\left\|\nabla_{A_y} \bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y})\right\|_F^2 = \sum_{k=1}^{d} \left\|\nabla_{(A_y)_k} \bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y})\right\|^2 \\
&= \sum_{k=1}^{d} \Bigg\| \frac{1}{n_y} \sum_{i=1}^{n_y} \sum_{j=1}^{N} \lambda(t_j)(t_j - t_{j-1})(\bar{\sigma}_{t_j} s_{t_j,\theta_y}(x(t_j)) + \xi_{ij})_k \sigma(W_y x(t_j) + U_y e(t_j)) \\
&\quad + \beta\omega(t_j)(t_j - t_{j-1}) \mathbb{E}_{y'}[(s_{t_j,\theta_y}(x(t_j)) - s_{t_j,\theta_{y'}}(x(t_j)))_k \sigma(W_y x(t_j) + U_y e(t_j))] \Bigg\|^2 \\
&\leq 2 \sum_{k=1}^{d} \left\| \frac{1}{n_y} \sum_{i=1}^{n_y} \sum_{j=1}^{N} \lambda(t_j)(t_j - t_{j-1})(\bar{\sigma}_{t_j} s_{t_j,\theta_y}(x(t_j)) + \xi_{ij})_k \sigma(W_y x(t_j) + U_y e(t_j)) \right\|^2 \\
&\qquad + 2\beta^2 \sum_{k=1}^{d} \Bigg\| \frac{1}{n_y} \sum_{i=1}^{n_y} \sum_{j=1}^{N} \omega(t_j)(t_j - t_{j-1}) \mathbb{E}_{y'}[(s_{t_j,\theta_y}(x(t_j)) - s_{t_j,\theta_{y'}}(x(t_j)))_k \cdot \\
&\qquad\qquad \sigma(W_y x(t_j) + U_y e(t_j))] \Bigg\|^2 \\
&\leq 2 \frac{N}{n_y} \sum_{i=1}^{n_y} \sum_{j=1}^{N} \lambda(t_j)^2 (t_j - t_{j-1})^2 \sum_{k=1}^{d} \left\| \bar{\sigma} s_{t_j,\theta_y}(x(t_j)) + \xi_{ij} \right\|^2 \left\| \sigma(W_y x(t_j) + U_y e(t_j)) \right\|^2 \\
&\qquad + 2\beta^2 \frac{N}{n_y} \sum_{i=1}^{n_y} \sum_{j=1}^{N} \omega(t_j)^2 (t_j - t_{j-1})^2 \cdot \\
&\qquad \sum_{k=1}^{d} \mathbb{E}_{y'}[\left\| s_{t_j,\theta_y}(x(t_j)) - s_{t_j,\theta_{y'}}(x(t_j)) \right\|^2] \left\| \sigma(W_y x(t_j) + U_y e(t_j)) \right\|^2 \\
&\leq 4Nd \left\| \sigma(W_y x(t_j) + U_y e(t_j)) \right\|_2^2 \max_j \{\lambda(t_j)(t_j - t_{j-1})\bar{\sigma}_{t_j}, \beta\omega(t_j)(t_j - t_{j-1})\} \bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y}) \\
&\leq 4Nd^2 (C_{t_N,\delta} + C_{t_N,e})^2 C^2_{w_y,u_y} \max_j \{\lambda(t_j)(t_j - t_{j-1})\bar{\sigma}_{t_j}, \beta\omega(t_j)(t_j - t_{j-1})\} \bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y})
\end{aligned}
$$

Since $w.p.1 - n_y N\delta$ the empirical loss function $\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y})$ is bounded, $\left\|\nabla_{A_y} \bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y})\right\|_F^2$ is bounded with the same probability.

This also shows that for fixed $\theta_{-y}, (W_y, U_y)_{y\in\mathcal{Y}}, w.p.1 - n_y N\delta$, $\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y})$ is a Lipschitz function in θ_y with Lipschitz constant σ_y such that $\sigma_y^2 = 4C_1 Nd^2 (C_{t_N,\delta} + C_{t_N,e})^2 C^2_{w_y,u_y} \max_j \{\lambda(t_j)(t_j - t_{j-1})\bar{\sigma}_{t_j}, \beta\omega(t_j)(t_j - t_{j-1})\} \left(\sum_{j=1}^{N} \frac{\lambda(t_j)(t_j - t_{j-1})}{\bar{\sigma}_{t_j}} + \beta\omega(t_j)(t_j - t_{j-1}) \right)$

B.5 Smoothness of Loss Function $\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y})$

Lemma 3. *Let $(W_y, U_y)_{y\in\mathcal{Y}}, \theta_{-y}, \{t_j\}_{j=1}^N$ be fixed. Let $L_y = d(C_{t_N,\delta} + C_{t_N,e})^2 C^2_{w_y,u_y} \sum_{j=1}^{N} \left(\lambda(t_j)(t_j - t_{j-1})\bar{\sigma}_{t_j} + \beta\omega(t_j)(t_j - t_{j-1}) \right)$. Then for $\delta > 0, \delta \ll 1$, w.p. $1 - n_y N\delta$, $\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y})$ is $\frac{L_y}{m^2}$ smooth and convex in θ_y.*

Proof We have,

$$
\begin{aligned}
\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y}) =& \frac{1}{2n_y} \sum_{i=1}^{n_y} \sum_{j=1}^{N} \frac{\lambda(t_j)(t_j - t_{j-1})}{\bar{\sigma}_{t_j}} \Bigg[\left\| \bar{\sigma}_{t_j} s_{t_j,\theta_y}(x(t_j)) + \xi_{ij} \right\|_2^2 \\
&+ \beta\omega(t_j)(t_j - t_{j-1}) \mathbb{E}_{y'\sim Q} \left[\left\| s_{t_j,\theta_y}(x(t_j)) - s_{t,\theta_{y'}}(x(t_j)) \right\|_2^2 \right] \Bigg]
\end{aligned} \tag{46}
$$

To show smoothness, we will show that the function $f(\theta_y) = \left\|\bar{\sigma}_{t_j} s_{t_j,\theta_y}(x(t_j)) + \xi_{ij}\right\|_2^2$ and $g(\theta_y) = \left\|s_{t_j,\theta_y}(x_{t_j}) - s_{t_j,\theta_{y'}}(x(t_j))\right\|_2^2$ are individually smooth. Once we prove this, it is easy to show $\mathcal{L}_{reg}^{n_y}(\theta_y, \theta_{-y})$ is smooth as the linear combination of smooth functions is again smooth. To show smoothness, we need to show that $\left\|\nabla^2_{\theta_y} f(\theta_y)\right\|$ and $\left\|\nabla^2_{\theta_y} g(\theta_y)\right\|$ have a bounded norm. Recall that $s_{t,\theta_y}(x) = \frac{1}{m} A_y \sigma(W_y x(t) + U_y e(t)$. Let $h_1(x,t) := \sigma(W_y x + U_y e(t)), h_2(x,t) := s_{t,\theta_{y'}}(x), h_3(i,j) = \xi_{ij}$, we have

$$
\begin{aligned}
f(\theta_y) =& \left\|\bar{\sigma}_{t_j} s_{t_j,\theta_y}(x(t_j)) + \xi_{ij}\right\|^2 && (47)\\
=& \frac{\bar{\sigma}_{t_j}^2}{m^2} h_1^T(x(t_j),t_j) A_y^T A_y h_1(x(t_j),t_j) - 2\bar{\sigma}_{t_j} h_3^T(i,j)\left(\frac{A_y}{m}\right) h_1(x(t_j),t_j) && (48)\\
&+ h_3^T(i,j) h_3(i,j) && (49)\\
\stackrel{a}{=}& \frac{\bar{\sigma}_{t_j}^2}{m^2} \text{trace}(A_y^T A_y B_1) - \frac{2\bar{\sigma}_{t_j}}{m} \text{trace}(A_y B_3) + \text{constant} && (50)\\
\stackrel{b}{=}& \frac{\bar{\sigma}_{t_j}^2}{m^2} vec(A_y)^T (B_1 \bigotimes I) vec(A_y) - \frac{2\bar{\sigma}_{t_j}}{m} vec(B_3^T)^T vec(A_y) + \text{constant} && (51)
\end{aligned}
$$

where (a) follows from the identity $x^T A y = \text{trace}(Byx^T)$, (b) follows from the following identities

$$
\begin{aligned}
\text{trace}(A^T AB) =& \text{trace}(ABA^T) = vec(A)^T (B \bigotimes I) vec(A)\\
\text{trace}(AB) =& vec(A)^T vec(B^T)
\end{aligned}
$$

and $B_3 = h_1(x(t_j),t_j) h_3^T(i,j)$.

Similarly, we have for $g(\theta_y)$

$$
\begin{aligned}
g(\theta_y) =& \left\|s_{t,\theta_y}(x(t_j)) - s_{t,\theta_{y'}}(x(t_j))\right\|^2\\
=& \frac{1}{m^2} h_1^T(x(t_j),t_j) A_y^T A_y h_1(x(t_j),t_j) - 2h_2^T(x(t_j),t_j)\left(\frac{A_y}{m}\right) h_1(x(t_j),t_j)\\
&+ h_2^T(x(t_j),t_j) h_2(x(t_j),t_j)\\
\stackrel{a}{=}& \frac{1}{m} \text{trace}(A_y^T A_y B_1) - \frac{2}{m} \text{trace}(A_y B_2) + \text{constant}\\
\stackrel{b}{=}& \frac{1}{m^2} vec(A_y)^T (B_1 \bigotimes I) vec(A_y) - \frac{2}{m} vec(B_2^T)^T vec(A) + \text{constant}
\end{aligned}
$$

where $B_1 := h_1(x(t_j),t_j) h_1^T(x(t_j),t_j)$ and $B_2 := h_1(x(t_j),t_j) h_2^T(x(t_j),t_j)$. Thus,

$$
\frac{1}{\bar{\sigma}_{t_j}^2} \nabla^2_{\theta_y} f(\theta_y) = \nabla^2_{\theta_y} g(\theta_y) = \nabla^2_{vec(A_y)} g(\theta_y) = \frac{2}{m^2}(B_1 \bigotimes I) \tag{52}
$$

The eigenvalues of $(B_1 \bigotimes I)$ is the same as B_1 with multiplicity. Thus, to show smoothness, we need to bound the maximum eigenvalues of B_1. For any $v \in \mathbb{R}^m$

$$
0 \leq v^T B_1 v = (v^T h_1(x(t_j),t_j))^2 \leq d \left\|h_1(x(t_j),t_j)\right\|_\infty^2 v^T v \tag{53}
$$

Now,

$$
\begin{aligned}
\left\|\sigma(W_y x(t_j) + U_y e(t_j))\right\|_\infty =& \max_{i=1,\cdots,m} \sigma(w_{y,i}^T x(t_j) + u_{y,i}^T e(t_j)) && (54)\\
\leq& \max_{i=1,\cdots,m} |w_{y,i}^T x(t_j) + u_{y,i}^T e(t_j)| && (55)\\
\leq& \max_{i=1,\cdots,m} \left\|w_{y,i}\right\|_1 \left\|x(t_j)\right\|_\infty + \left\|u_{y,i}\right\|_1 \left\|e(t_j)\right\|_\infty && (56)\\
\leq& (C_{t_N,\delta} + C_{t_N,e}) C_{w_y,u_y} && (57)
\end{aligned}
$$

Thus, we have for any $v \in \mathbb{R}^m$

$$0 \le v^T B_1 v \le d(C_{t_N,\delta} + C_{t_N,e})^2 C^2_{w_y,u_y} v^T v \tag{58}$$

Since $w.p.\ 1 - n_y N\delta$ we have $\{\|x_{ij}\|_\infty \le C_{t_N,\delta}\}_{i=1,j=1}^{n_y,N}$, we have with the same probability $f(\theta_y)$ and $g(\theta_y)$ are smooth in θ_y for every $W_y, U_y, x(t_j), \theta_{-y}$.

Thus, $\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y})$ is $L_y \frac{1}{m^2} = \frac{1}{m^2} d(C_{t_N,\delta} + C_{t_N,e})^2 C^2_{w_y,u_y} \sum_{j=1}^N \bigg(\lambda(t_j)(t_j - t_{j-1})\bar{\sigma}_{t_j} + \beta\omega(t_j)(t_j - t_{j-1})\bigg)$ smooth.

B.6 Proof: First order convergence of the algorithm

Proof Our proof follows closely along the lines of [11]. Let $\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y})$ be the empirical version of $\mathcal{L}^{y}_{reg}(\theta_y, \theta_{-y})$ with n_y samples. By L_y smoothness of $\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y})$ we have, for any $y \in \mathcal{Y}$,

$$\begin{aligned} \bar{\mathcal{L}}^{n_y}_{reg}(\theta^{\tau+1}_y, \theta^\tau_{-y}) &\le \bar{\mathcal{L}}^{n_y}_{reg}(\theta^\tau_y, \theta^\tau_{-y}) + \langle \nabla_{\theta_y} \bar{\mathcal{L}}^{n_y}_{reg}(\theta^\tau_y, \theta^\tau_{-y}), \theta^{\tau+1}_y - \theta^\tau_y \rangle \\ &\quad + \frac{L_y}{2} \left\|\theta^{\tau+1}_y - \theta^\tau_y\right\|^2 \end{aligned} \tag{59}$$

$$\begin{aligned} \implies \bar{\mathcal{L}}^{n_y}_{reg}(\theta^{\tau+1}_y, \theta^\tau_{-y}) &\le \bar{\mathcal{L}}^{n_y}_{reg}(\theta^\tau_y, \theta^\tau_{-y}) - \eta_\tau \left\|\nabla_{\theta_y} \bar{\mathcal{L}}^{n_y}_{reg}(\theta^\tau_y, \theta^\tau_{-y})\right\|^2 \\ &\quad + \frac{L_y}{2}\eta_\tau^2 \left\|\nabla_{\theta_y} \bar{\mathcal{L}}^{n_y}_{reg}(\theta^\tau_y, \theta^\tau_{-y})\right\|^2 \end{aligned} \tag{60}$$

$$\begin{aligned} \implies \eta_\tau \left\|\nabla_{\theta_y} \bar{\mathcal{L}}^{n_y}_{reg}(\theta^\tau_y, \theta^\tau_{-y})\right\|^2 &\le \bar{\mathcal{L}}^{n_y}_{reg}(\theta^\tau_y, \theta^\tau_{-y}) - \bar{\mathcal{L}}^{n_y}_{reg}(\theta^{\tau+1}_y, \theta^\tau_{-y}) \\ &\quad + \frac{L_y}{2}\eta_\tau^2 \left\|\nabla_{\theta_y} \bar{\mathcal{L}}^{n_y}_{reg}(\theta^t_y, \theta^t_{-y})\right\|^2 \end{aligned} \tag{61}$$

$$\begin{aligned} \implies \sum_{\tau=1}^{T_{train}} \eta_\tau \left\|\nabla_{\theta_y} \bar{\mathcal{L}}^{n_y}_{reg}(\theta^\tau_y, \theta^\tau_{-y})\right\|^2 &\le \bar{\mathcal{L}}^{n_y}(\theta^t_y) - \bar{\mathcal{L}}^{n_y}(\theta^{\tau+1}_y) + \beta \sum_{\tau=1}^{T_{train}} \psi(\theta^{\tau+1}_y, \theta^\tau_y, \theta^\tau_{-y}) \\ &\quad + \sum_{\tau=1}^{T_{train}} \frac{L_y}{2}\eta_\tau^2 \left\|\nabla_{\theta_y} \bar{\mathcal{L}}^{n_y}_{reg}(\theta^\tau_y, \theta^\tau_{-y})\right\|^2 \end{aligned} \tag{62}$$

$$\begin{aligned} \implies \sum_{\tau=1}^{T_{train}} \eta_\tau \left\|\nabla_{\theta_y} \bar{\mathcal{L}}^{n_y}_{reg}(\theta^\tau_y, \theta^\tau_{-y})\right\|^2 &\le \bar{\mathcal{L}}^{n_y}(\theta^1_y) - \bar{\mathcal{L}}^{n_y}(\theta^{\tau+1}_y) + \sum_{\tau=1}^{T_{train}} \frac{L_y}{2}\eta_\tau^2 \sigma_y^2 \\ &\quad + \beta \sum_{\tau=1}^{T_{train}} \psi(\theta^{\tau+1}_y, \theta^\tau_y, \theta^\tau_{-y}) \end{aligned} \tag{63}$$

$$\begin{aligned} \implies \min_{\tau \in [T_{train}]} \left\|\nabla_{\theta_y} \bar{\mathcal{L}}^{n_y}_{reg}(\theta^\tau_y, \theta^\tau_{-y})\right\|^2 &\le \frac{\bar{\mathcal{L}}^{n_y}(\theta^1_y) - \bar{\mathcal{L}}^{n_y}(\theta^*_y) + \frac{L_y}{2}\sigma_y^2 \sum_{\tau=1}^{T_{train}} \eta_\tau^2}{\sum_{\tau=1}^{T_{train}} \eta_\tau} \\ &\quad + \beta \frac{\sum_{t=1}^{T_{train}} \psi(\theta^{\tau+1}_y, \theta^\tau_y, \theta^\tau_{-y})}{\sum_{\tau=1}^{T} \eta_\tau} \end{aligned} \tag{64}$$

where $\psi(\theta^{\tau+1}_y, \theta^\tau_y, \theta^\tau_{-y}) = \bar{\mathcal{L}}^{n_y}_{mut}(\theta^\tau_y, \theta^\tau_{-y}) - \bar{\mathcal{L}}^{n_y}_{mut}(\theta^{\tau+1}_y, \theta^\tau_{-y})$.

B.6.1 Analyzing the Bias Term

Lemma 4. *Suppose $\theta_{-y}, (W_y, U_y)_{y\in\mathcal{Y}}$ are fixed. Let $\phi_y = d^{1.5} N (C_{t_N,\delta} + C_{t_N,e})^2 C^2_{w_y,u_y} B \max_j \omega(t_j)(t_j - t_{j-1})$. Then for $\delta > 0, \delta \ll 1, w.p. 1 - n_y N\delta$, we have*

$$\bar{\mathcal{L}}^{n_y}_{mut}(\theta_y, \theta_{-y}) = \frac{1}{2n_y} \sum_{i=1}^{n_y} \sum_{j=1}^{N} \omega(t_j)(t_j - t_{j-1}) \mathbb{E}_{y' \sim Q}\left[\left\|s_{t_j,\theta_y}(x(t_j)) - s_{t,\theta_{y'}}(x(t_j))\right\|_2^2\right] \tag{65}$$

is ϕ_y Lipschitz in θ_y.

Proof:

$$
\begin{aligned}
&\left\|\nabla_{A_y}\bar{\mathcal{L}}^{n_y}_{reg,mut}(\theta_y,\theta_{-y})\right\|_F^2 = \sum_{k=1}^{d}\left\|\nabla_{(A_y)_k}\bar{\mathcal{L}}^{n_y}_{mut}(\theta_y,\theta_{-y})\right\|^2 \\
&= \sum_{k=1}^{d}\left\|\frac{1}{n_y}\sum_{i=1}^{n_y}\sum_{j=1}^{N}\omega(t_j)(t_j-t_{j-1})\mathbb{E}_{y'}[(s_{t_j,\theta_y}(x(t_j))-s_{t_j,\theta_{y'}}(x(t_j)))_k\sigma(W_yx(t_j)+U_ye(t_j))]\right\|^2 \\
&\qquad\qquad\qquad \le \frac{N}{n_y}\sum_{i=1}^{n_y}\sum_{j=1}^{N}\omega(t_j)^2(t_j-t_{j-1})^2.
\end{aligned}
$$

$$
\begin{aligned}
&\sum_{k=1}^{d}\mathbb{E}_{y'}[\left\|s_{t_j,\theta_y}(x(t_j))-s_{t_j,\theta_{y'}}(x(t_j))\right\|^2]\left\|\sigma(W_yx(t_j)+U_ye(t_j))\right\|^2 \\
&\le \left\|\sigma(W_yx(t_j)+U_ye(t_j))\right\|^2 Nd\max_j\omega(t_j)(t_j-t_{j-1})\bar{\mathcal{L}}^{n_y}_{mut}(\theta_y,\theta_{-y}) \\
&\le 4d^2N(C_{t_N,\delta}+C_{t_N,e})^2C^2_{w_y,u_y}\max_j\omega(t_j)(t_j-t_{j-1})\bar{\mathcal{L}}^{n_y}_{mut}(\theta_y,\theta_{-y}) \\
&\quad \le d^3N(C_{t_N,\delta}+C_{t_N,e})^4C^4_{w_y,u_y}B^2\max_j\omega(t_j)(t_j-t_{j-1})\sum_{j=1}^{N}\omega(t_j)(t_j-t_{j-1})
\end{aligned}
$$

Since $\bar{\mathcal{L}}^{n_y}_{mut}(\theta_y,\theta_{-y}) \le \bar{\mathcal{L}}^{n_y}_{reg}(\theta_y,\theta_{-y})$ and $w.p.1-n_yN\delta$, $\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y,\theta_{-y})$ is bounded. Thus, $\left\|\nabla_{A_y}\bar{\mathcal{L}}^{n_y}_{mut}(\theta_y,\theta_{-y})\right\|_F^2$ is bounded and hence $\bar{\mathcal{L}}^{n_y}_{mut}(\theta_y,\theta_{-y})$ is Lipschitz in θ_y with $\phi_y = d^{1.5}N(C_{t_N,\delta}+C_{t_N,e})^2C^2_{w_y,u_y}B\max_j\omega(t_j)(t_j-t_{j-1})$ Here,

$$
\begin{aligned}
\psi(\theta_y^{\tau+1},\theta_y^{\tau},\theta_{-y}^{\tau}) =&\left|\bar{\mathcal{L}}^{n_y}_{mut}(\theta_y^{\tau},\theta_{-y}^{t})-\bar{\mathcal{L}}^{n_y}_{mut}(\theta_y^{\tau+1},\theta_{-y}^{\tau})\right| && (66)\\
\le&\phi_y\left\|\theta_y^{\tau}-\theta_y^{\tau+1}\right\| && (67)\\
\le&\phi_y\eta_t\left\|\nabla_{A_y}\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y^{\tau},\theta_{-y}^{\tau})\right\| && (68)\\
\le&\phi_y\eta_\tau\sigma_y && (69)
\end{aligned}
$$

By taking $\eta_\tau \le \frac{m^2}{\max_{y\in\mathcal{Y}}L_y\sqrt{T_{train}}}, \forall y\in\mathcal{Y}$

$$
\max_{y\in\mathcal{Y}}\min_{\tau\in[T_{train}]}\left\|\nabla_{\theta_y}\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y^{\tau},\theta_{-y}^{\tau})\right\|^2 = \max_{y\in\mathcal{Y}}\mathcal{O}\left(\frac{2(\bar{\mathcal{L}}^{n_y}(\theta_y^0))-\bar{\mathcal{L}}^{n_y}(\theta_y^*))}{\max_{y\in\mathcal{Y}}L_y\sqrt{T_{train}}}+\frac{\sigma_y^2}{\sqrt{T_{train}}}+\beta\phi_y\sigma_y\right) \quad (70)
$$

$$
= \mathcal{O}\left(\frac{m^2}{\sqrt{T_{train}}}+\beta\right) \quad (71)
$$

For (θ_y,θ_{-y}), we have

$$
\begin{aligned}
\text{NE-gap}(\theta_y,\theta_{-y}) &= \max_{y\in Y}\mid\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y,\theta_{-y})-\bar{\mathcal{L}}^{n_y}_{reg}(B(\theta_{-y}),\theta_{-y})\mid && (72)\\
&\le \max_{y\in\mathcal{Y}}\left\|\nabla_{\theta_y}\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y,\theta_{-y})\right\|^2\left\|\theta_y-B(\theta_{-y})\right\|_2^2 && (73)
\end{aligned}
$$

Since the strategy space for θ_y is bounded in norm. We have

$$
\text{NE-gap}(\theta_y,\theta_{-y}) \lesssim \max_{y\in\mathcal{Y}}\left\|\nabla_{\theta_y}\bar{\mathcal{L}}^{n_y}_{reg}(\theta_y,\theta_{-y})\right\|^2 \quad (74)
$$

$$
\implies \min_{\tau\in[T_{train}]}\text{NE-gap}(\theta_y^{\tau},\theta_{-y}^{\tau}) = \mathcal{O}\left(\frac{m^2}{\sqrt{T_{train}}}+\beta\right) \quad (75)
$$

B.7 Monte Carlo Error of the Finite Neural Network

Observe that

$$\mathcal{L}^y(\theta_y) = \frac{1}{2}\sum_{j=1}^{N}\lambda(t_j)(t_j - t_{j-1})\mathbb{E}_{X_0}\mathbb{E}_{X_{t_j}|X_0}\left[\left\|\nabla_{x(t_j)}\log p_t(x_i(t_j)|x_0) - s_{t_j,\theta_y}(x_i(t_j))\right\|_2^2\right] \tag{76}$$

$$+\frac{1}{2}\sum_{j=1}^{N}\lambda(t_j)(t_j - t_{j-1})C_{t_j} \tag{77}$$

$$= \frac{1}{2}\sum_{j=1}^{N}\lambda(t_j)(t_j - t_{j-1})\mathbb{E}_{x(t_j)\sim p_{t_j}}\left[\left\|s_{t_j,\theta_y}(x(t_j)) - \nabla_x \log p_{t_j}(x(t_j))\right\|_2^2\right]\Bigg] \tag{78}$$

For each $y \in \mathcal{Y}$, $\mathcal{L}^y(\theta_y^*)$ is the optimal loss function for the unregularized version under the current hypothesis class. Let $\mathcal{L}^y(\bar{\theta}_y^*)$ be the optimal unregularized loss function under the continuous version of the random feature model. Then,

$$\mathcal{L}^y(\theta_y^*) = \frac{1}{2}\sum_{j=1}^{N}\lambda(t_j)(t_j - t_{j-1})\mathbb{E}_{x(t_j)\sim p_{t_j}}\left[\left\|s_{t_j,\theta_y^*}(x(t_j)) - \nabla_x \log p_{t_j}(x(t_j))\right\|_2^2\right]\Bigg] \tag{79}$$

$$\leq 2\Bigg(\frac{1}{2}\sum_{j=1}^{N}\lambda(t_j)(t_j - t_{j-1})\mathbb{E}_{x(t_j)\sim p_{t_j}}\left[\left\|\bar{s}_{t,\bar{\theta}_y^*}(x(t)) - \nabla_x \log p_t(x(t))\right\|_2^2\right] \tag{80}$$

$$+\frac{1}{2}\sum_{j=1}^{N}\lambda(t_j)(t_j - t_{j-1})\mathbb{E}_{x(t_j)\sim p_{t_j}}\left[\left\|\bar{s}_{t_j,\bar{\theta}_y^*}(x(t_j)) - s_{t_j,\theta_y^*}(x(t_j))\right\|_2^2\right]\Bigg) \tag{81}$$

$$\leq 2\mathcal{L}^y(\bar{\theta}_y^*) + Err_{MC}(\theta_y^*, \bar{\theta}_y^*; \{t_j\}_{j=1}^N, \{\lambda(t_j)\}_{j=1}^N) \tag{82}$$

Proposition 2. ***Monte Carlo estimates***. *Define the Monte Carlo error*

$$Err_{MC}(\theta, \bar{\theta}, \{t_j\}_{j=1}^N, \{\lambda(t_j)\}_{j=1}^N) := \sum_{j=1}^{N}\lambda(t_j)(t_j - t_{j-1}). \\ \mathbb{E}_{x(t_j)\sim p_{t_j}}\left[\left\|\bar{s}_{t_j,\bar{\theta}_y^*}(x(t_j)) - s_{t_j,\theta_y^*}(x(t_j))\right\|_2^2\right] \tag{83}$$

Suppose that $\|X(0)\|_\infty \leq K$ and the trainable parameter a and embedding functions $W, U, e(.)$ are both bounded. Then. given any $\bar{\theta}$. for any $\delta > 0, \delta \ll 1$, with probability of at least $1 - 2N\delta$, there exists θ such that

$$Err_{MC}(\theta, \bar{\theta}, \{t_j\}_{j=1}^N, \{\lambda(t_j)\}_{j=1}^N) \leq \frac{2C_{w,u}^2 B^2 (C_{t_N,\delta} + C_{t_N.e})^2 d^2}{m}\log\left(\frac{2}{\delta}\right)\sum_{j=1}^{N}\lambda(t_j)(t_j - t_{j-1}) \tag{84}$$

Proof. The proof closely along the line of [15]. Fix any $\bar{\theta}$. For notational convenience, we will drop y from θ_y and $\bar{\theta}_y$. For $k = 1, 2, \cdots d$, define

$$Z_{t,k}(W, U) := \left\|s_{t,\theta,k}(x) - \bar{s}_{t,\bar{\theta},k}(x)\right\|_{L^2(p_t)} = \mathbb{E}_{x\sim p_t}^{1/2}\left[|s_{t,\theta,k}(x) - \bar{s}_{t,\bar{\theta},k}(x)|^2\right] \tag{85}$$

$$= \mathbb{E}_{x\sim p_t}\left[\left|\frac{1}{m}\sum_{i=1}^{m}a_{i,k}\sigma(w_i^T x + u_i^T e(t)) - \mathbb{E}_{(w,u)}[a_k(w,u)\sigma(w^T x + u^T e(t))]\right|^2\right] \tag{86}$$

Then, we have

$$\mathbb{E}_{x\sim p_t}\left[\left\|s_{t,\theta_y}(x)-\bar{s}_{t,\bar{\theta}_y}(x)\right\|_2^2\right]=\sum_{k=1}^{d}\mathbb{E}_{x\sim p_t}\left[|s_{t,\theta_y,k}(x)-\bar{s}_{t,\bar{\theta}_y,k}|^2\right] \tag{87}$$

$$=\sum_{k=1}^{d}Z_{t,k}^2(W,U) \tag{88}$$

$$\leq\sum_{k=1}^{d}\Big(|Z_{t,k}(W,U)-\mathbb{E}_{W,U}[Z_{t,k}]|+|\mathbb{E}_{W,U}[Z_{t,k}(W,U)]|\Big)^2 \tag{89}$$

$$\overset{(a)}{\leq}2\sum_{k=1}^{d}\Big(|Z_{t,k}(W,U)-\mathbb{E}_{W,U}[Z_{t,k}(W,U)]|^2 \tag{90}$$

$$+\mathbb{E}_{W,U}[Z_{t,k}^2(W,U)]\Big) \tag{91}$$

where (a) follows from the fact that $(a+b)^2\leq 2(a^2+b^2)$ and Jensen's Inequality $\mathbb{E}^2[Z_{t,k}(W,U)]\leq\mathbb{E}_{W,U}[Z_{t,k}^2(W,U)]$. According to Lemma 1. for any $\delta>0,\delta\ll 1$, w.p. atleast $1-\delta$, we have

$$\|x(t)\|_\infty\leq C_{t_N,\delta} \tag{92}$$

If $(\tilde{W},\tilde{U})$ is different from (W,U) at only one component indexed by i, we have w.p. $1-\delta$

$$|Z_{t,k}(W,U)-Z_{t,k}(\tilde{W},\tilde{U})| \tag{93}$$

$$=\left|\left\|s_{t,\theta,k}(x)-\bar{s}_{t,\bar{\theta},k}(x)\right\|_{L^2(p_t)}-\left\|s_{t,\tilde{\theta},k}(x)-\bar{s}_{t,\bar{\theta},k}(x)\right\|_{L^2(p_t)}\right| \tag{94}$$

$$\overset{(a)}{\leq}\left\|s_{t,\tilde{\theta},k}(x)-s_{t,\theta,k}(x)\right\|_{L^2(p_t)} \tag{95}$$

$$=\frac{1}{m}\left\|a_{i,k}\sigma(w_i^Tx+u_i^Te(t))-\tilde{a}_{i,k}\sigma(\tilde{w}_i^Tx+\tilde{u}_i^Te(t))\right\|_{L^2(p_t)} \tag{96}$$

$$\overset{(b)}{\leq}\frac{1}{m}\left(|a_{i,k}|\left\|\sigma(w_i^Tx+u_i^Te(t))\right\|_{L^2(p_t)}+|\tilde{a}_{i,k}|\left\|\sigma(\tilde{w}_i^Tx+\tilde{u}_i^Te(t))\right\|_{L^2(p_t)}\right) \tag{97}$$

$$\overset{(c)}{\leq}\frac{1}{m}\left(|a_{i,k}|\left\|w_i^Tx+u_i^Te(t))\right\|_{L^2(p_t)}+|\tilde{a}_{i,k}|\left\|(\tilde{w}_i^Tx+\tilde{u}_i^Te(t))\right\|_{L^2(p_t)}\right) \tag{98}$$

$$\overset{(d)}{\leq}\frac{1}{m}\left(|a_{i,k}|(\|w_i\|_1C_{t_N,\delta}+\|u_i\|_1\|e(t)\|_\infty)+|\tilde{a}_{i,k}|(\|\tilde{w}_i\|_1C_{t_N,\delta}+\|\tilde{u}_i\|_1\|e(t)\|_\infty)\right) \tag{99}$$

$$\overset{(e)}{\leq}\frac{2}{m}BC_{w,u}(C_{t_N,\delta}+C_{t_N,e}) \tag{100}$$

where (a) and (b) follows from triangle inequality $|\,\|a\|-\|b\|\,|\leq\|a-b\|$ and $\|a-b\|\leq\|a\|+\|b\|$, (c) follows from the fact that $|\sigma(y)|\leq|y|$, (d) follows from Lemma 1 and Holder Inequality, (e) follows from the bounds on $\|w_i\|_1,\|u_i\|_1,x,|a_{i,k}|,e(t_j)$.

Thus, $w,p.1-\delta$, $Z_{t,k}(W,U)$ has bounded increment property. Using McDiarmid's inequality, $w.p.1-2\delta$, we have

$$|Z_{t,k}(W,U)-\mathbb{E}_{W,U}[Z_{t,k}(W,U)]|\leq\frac{B}{m}C_{w,u}(C_{t_N,\delta}+C_{t_N.e})\sqrt{d\log(\frac{2}{\delta})} \tag{101}$$

Now we compute

$$
\begin{aligned}
&\mathbb{E}_{W,U}[Z_{t,k}^2(W,U)] \\
=&\mathbb{E}_{W,U}\left[\mathbb{E}_{x\sim p_t}[|s_{t,\theta,k}(x) - \bar{s}_{t,\bar{\theta},k}(x)|^2]\right] \\
\overset{(b)}{=}&\mathbb{E}_{x\sim p_t}\left[\mathbb{E}_{W,U}[|s_{t,\theta,k}(x) - \bar{s}_{t,\bar{\theta},k}(x)|^2]\right] \\
=&\frac{1}{m^2}\mathbb{E}_{x\sim p_t}\left[\mathbb{E}_{W,U}\left[\left|\sum_{i=1}^m \left(a_{i,k}\sigma(w_i^T x + u_i^T e(t)) - \mathbb{E}_{w,u}[a_k(w,u)\sigma(w^T x + u^T e(t))]\right)\right|^2\right]\right] \\
&+ \frac{1}{m^2}\mathbb{E}_{x\sim p_t}\left[\mathbb{E}_{W,U}\left[\sum_{i\neq j}(a_{i,k}\sigma(w_i^T x + u_i^T e(t)) - \mathbb{E}_{w,u}[a_k(w,u)\sigma(w^T x + u^T e(t))])\right.\right. \\
&\left.\left.\times (a_{j,k}\sigma(w_j^T x + u_j^T e(t)) - \mathbb{E}_{w,u}[a_k\sigma(w^T x + u^T e(t))])\right]\right] \\
\overset{(c)}{=}&\frac{1}{m^2}\mathbb{E}_{x\sim p_t}\left[\mathbb{E}_{W,U}\left[\sum_{i=1}^m \left(a_{i,k}\sigma(w_i^T x + u_i^T e(t)) - \mathbb{E}_{w,u}[a_k(w,u)\sigma(w^T x + u^T e(t))]\right)^2\right]\right] \\
&+ \frac{1}{m^2}\mathbb{E}_{x\sim p_t}\left[\sum_{i\neq j}\mathbb{E}_{w_i,u_i}\left[(a_{i,k}\sigma(w_i^T x + u_i^T e(t)) - \mathbb{E}_{w,u}[a_k(w,u)\sigma(w^T x + Ue(t))])\right.\right. \\
&\left.\left.\times \mathbb{E}_{w_j,u_j}\left[(a_{j,k}\sigma(w_j^T x + u_j^T e(t)) - \mathbb{E}_{w,u}[a_k(w,u)\sigma(w^T x + Ue(t))])\right]\right]\right] \\
\overset{(d)}{=}&\frac{1}{m^2}\mathbb{E}_{x\sim p_t}\left[\sum_{i=1}^m \mathbb{E}_{W,U}\left[\left(a_{i,k}\sigma(w_i^T x + u_i^T e(t)) - \mathbb{E}_{w,u}[a_k(w,u)\sigma(w^T x + u^T e(t))]\right)^2\right]\right] \\
\overset{(e)}{\leq}&\frac{1}{m}\mathbb{E}_{x\sim p_t}\left[\mathbb{E}_{w,u}\left[(a_k(w,u)\sigma(w^T x + u^T e(t)))^2\right]\right] \\
\overset{(f)}{\leq}&\frac{1}{m}\mathbb{E}_{x\sim p_t}\left[\mathbb{E}_{w,u}\left[(|a_{y,k}(w,u)|(\|w\|_1 C_{t_N,\delta} + \|u\|_1 \|e(t)\|_\infty))^2\right]\right] \\
\leq&\frac{1}{m}(C_{t_N,\delta} + C_{t_N,e})^2 C_{w,u}^2 B^2
\end{aligned}
$$

where (b) is due to Fubini's theorem, (c) is due to independence of sampling (w_i, u_i) and (w_j, u_j), (d) is due to $a_{j,k}\sigma(w_j^T x + u_j^T e(t))$ being an unbiased estimator of the continuous version of score network, (e) follows from $Var(X) \leq \mathbb{E}[X^2]$, (f) follows from $|\sigma(y)| \leq |y|$ and Holder's inequality. Thus. $w.p.1 - 2\delta$,

$$\mathbb{E}_{x\sim p_t}\left[\left\|s_{t,\theta_y}(x) - \bar{s}_{t,\bar{\theta}_y}(x)\right\|_2^2\right] \leq \frac{2C_{w,u}^2 B^2 (C_{t_N,\delta} + C_{t_N.e})^2 d^2}{m}\log\left(\frac{2}{\delta}\right) \tag{102}$$

Finally, we have $w.p.1 - 2N\delta$

$$Err_{MC}(\theta, \bar{\theta}, \{t_j\}_{j=1}^N, \{\lambda(t_j)\}_{j=1}^N) \leq \frac{2C_{w,u}^2 B^2 (C_{t_N,\delta} + C_{t_N.e})^2 d^2}{m}\log\left(\frac{2}{\delta}\right)\sum_{j=1}^N \lambda(t_j)(t_j - t_{j-1}) \tag{103}$$

B.8 Radamacher Complexity

In this section, we will bound the term related to the generalization bound

$$\sup_{(\theta_y,\theta_{-y})} |\ \mathcal{L}_{reg}^y(\theta_y, \theta_{-y}) - \bar{\mathcal{L}}_{reg}^{n_y}(\theta_y, \theta_{-y})\ | \tag{104}$$

The Rademacher complexity of a real valued function class $\mathcal{F}$ is defined as:

$$\mathcal{R}_n(\mathcal{F}) := \mathbb{E}_{x_1,\cdots,x_n}\mathbb{E}_{\sigma_1,\cdots,\sigma_n}\left[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n \sigma_i f(x_i)\right], \tag{105}$$

The variables $\sigma_1, \cdots, \sigma_m$ are iid Bernoulli random variables that take values $\{+1, -1\}$ with equal probability and are independent of $x_1, \cdots, x_m$. However, for our random feature model, we have a vector valued function class

$$\hat{\mathcal{F}}_{W,U} := \left\{ f(x) = \frac{A}{m}\Phi(x, W, U) = \frac{1}{m}\sum_{k=1}^{m} \alpha_k \phi(x, w_k, u_k) \middle| \, \|A\|_F \leq B \right\} \tag{106}$$

Theorem 4. *[17, Theorem 3] Let X be nontrivial, symmetric and subgaussian. Then there exists a constant $C < \infty$, depending only on the distribution of X, such that for any countable set S and functions $\psi_i : S \to \mathbb{R}, \phi_i : S \to l_2, 1 \leq i \leq n$ satisfying*

$$\forall s, s' \in S, \psi_i(s) - \psi(s'_i) \leq \|\phi_i(s) - \phi_i(s')\| \tag{107}$$

we have

$$\mathbb{E} \sup_{s \in S} \sum_i \epsilon_i \psi_i(s) \leq C \mathbb{E} \sup_{s \in S} \sum_{i,k} X_{ik} \phi_i(s)_k \tag{108}$$

where the X_{ik} are independent copies of X for $1 \leq i \leq n$ and $1 \leq k \leq \infty$ and $\phi_i(s)_k$ is the k-th coordinate of $\phi_i(s)$. If X is a Rademacher variable we may choose $C = \sqrt{2}$, if X is a standard normal $C = \sqrt{\frac{\pi}{2}}$.

Corollary 2. *[17, Corollary 4] Let $\mathcal{X}$ be any set, $(x_1, \cdots, x_n) \in \mathcal{X}^n$, let F be a class of functions $f : \mathcal{X} \to l_2$ and let $h_i : l_2 \to \mathbb{R}$ have Lipschitz norm L. Then*

$$\mathbb{E} \sup_{f \in F} \sum_i \epsilon_i h_i(f(x_i)) \leq \sqrt{2} L \mathbb{E} \sup_{f \in F} \sum_{i,k} \epsilon_{ik} f_k(x_i) \tag{109}$$

where ϵ_{ik} is an independent doubly indexed Rademacher sequence and $f_k(x_i)$ is the k-th component of $f(x_i)$.

Lemma 5. *[17] Consider the function class $\mathcal{F} = \{x \to \frac{A}{m}\phi(x, W, U) : A \in \mathcal{B}(H, \mathbb{R}), \|A\|_F \leq B\}$. Then the empirical Rademacher complexity of F is*

$$\hat{Rad}_n(\mathcal{F}) = \mathbb{E} \sup_{f \in F} \sum_{i,k} \epsilon_{ik} f_k(x_i) \leq \frac{B}{\sqrt{m}} \sqrt{\sum_i \|\phi(x, W, U)\|^2} \tag{110}$$

Moreover, if $\mathbb{E}_x \|\phi(x, W, U)\|^2 \leq C^2$, the Rademacher Complexity of $\mathcal{F}$ is

$$\mathcal{R}_n(\mathcal{F}) \leq \frac{BC}{\sqrt{mn}} \tag{111}$$

Proof:

$$\hat{Rad}_n(\mathcal{F}) = \mathbb{E} \sup_{f \in F} \sum_{i,k} \epsilon_{ik} f_k(x_i) = \frac{1}{m} \mathbb{E} \sup_{\|A\|_F \leq B} \sum_k \langle a_k, \sum_i \epsilon_{ik} x_i \rangle \tag{112}$$

$$= \frac{1}{m} \mathbb{E} \sup_{\|A\|_F \leq B} tr(D^* A) \leq B \mathbb{E} \|D^*\|_* \tag{113}$$

where $D \in \mathcal{B}(H, \mathbb{R}^K)$ is the random transformation

$$v \to \left(\langle v, \sum_i \epsilon_{i1} x_i \rangle, \cdots, \langle v, \sum_i \epsilon_{iK} x_i \rangle \right) \tag{114}$$

Thus,

$$\mathbb{E} \|D^*\|_* = \mathbb{E} \sqrt{\sum_m \left\| \sum_i \epsilon_{ik} \phi(x_i, W, U) \right\|^2} \leq \sqrt{m \sum_i \|\phi(x_i, W, U)\|^2} \tag{115}$$

Thus,

$$\mathcal{R}_n(\mathcal{F}) = \mathbb{E}_{x_1,\cdots,x_n}\frac{1}{n}\hat{Rad}_n(\mathcal{F}) \leq \frac{B}{\sqrt{mn}}\mathbb{E}_{x_1,\cdots,x_n}\sqrt{\sum_i \|\phi(x_i,W,U)\|^2} \tag{116}$$

$$\leq \frac{B}{n\sqrt{m}}\sqrt{\sum_i \mathbb{E}_{x_1,\cdots,x_n}\|\phi(x_i,W,U)\|^2} \tag{117}$$

$$\leq \frac{BC}{\sqrt{mn}} \tag{118}$$

Suppose $0 < t_1 < \cdots < t_N = T$ are the chosen points of discretization for training, we have from the forward process

$$X(t) = \quad e^{-t}X(0) + \sqrt{1-e^{-2t}}Z, Z \sim N(0,1) \tag{119}$$

$$\implies \mathbb{E}_Z[X^2(t)] = \quad e^{-2t}x^2(0) + \frac{1-e^{-2t}}{2} \tag{120}$$

$$\implies \mathbb{E}_{X(0)}\mathbb{E}_Z[X^2(t_j)] \leq \quad K^2 + \frac{1-e^{-2T}}{2}, \forall 0 < t_j < T \tag{121}$$

Using the above bounds along with bounded support of embedding matrices W, U and embedding function $e(t)$ and Assumption 1, it is easy to show that

$$\mathbb{E}_{X_0}\mathbb{E}_{\xi_j}\left[\|\sigma(Wx(t)+Ue(t))\|_2^2\right] \leq F_T^2, \forall 0 < t \leq T \tag{122}$$

for some constant F_T^2 and $x(t) = e^{-t}x(0) + \sqrt{1-e^{-2t}}\xi_j, \xi_j \sim \mathcal{N}(0,I)$

Lemma 6. *The term*

$$\mathcal{L}^y_{mut}(\theta_y,\theta_{-y}) = \sum_{j=1}^N \omega(t_j)(t_j - t_{j-1})\mathbb{E}_{X_0}\mathbb{E}_{X_{t_j}|X_0}\mathbb{E}_{y'\sim Q}\left[\left\|s_{t_j,\theta_y}(x_i(t_j)) - s_{t,\theta_{y'}}(x_i(t_j))\right\|_2^2\right]\Bigg] \tag{123}$$

is $\mathcal{O}\Bigg(F_T B \sum_{j=1}^N \omega(t_j)(t_j - t_{j-1})\Bigg)$

Proof: Using the fact of bounded support of embedding matrices W, U and embedding function $e(t)$, bounded strategy space and Assumption 1 and eq 122, we get the desired bounded.

Lemma 7. *Suppose* $L_{C_1} = \bar{\sigma}^2_{t_j}BF_T + \sqrt{d}\sqrt{\log\frac{2}{\pi\delta^2}})$*. Then, with probability* $1-\delta$*, the function* $h : \mathcal{A} \subset \mathbb{R}^d \to \mathbb{R}$

$$h(x) = \left\|\bar{\sigma}_{t_j}x + \xi_{ij}\right\|^2 \tag{124}$$

is Lipschitz in x, where $\mathcal{A} = \{x \in \mathbb{R}^d : \|x\|_2 \leq F_T B\}$*.*

Proof. It is sufficient to show the norm of the gradient of $h(x)$ is bounded for $x \in \mathcal{A}$. With probability $1-\delta$,

$$\|\nabla_x h(x)\|_2 = \bar{\sigma}_{t_j}\left\|\bar{\sigma}_{t_j}x + \xi_{ij}\right\|_2 \leq \bar{\sigma}^2_{t_j}F_T B + \sqrt{d}\sqrt{\log\frac{2}{\pi\delta^2}}) \tag{125}$$

$$\tag{126}$$

Lemma 8. *Suppose* $L_{C_2} = 2F_T B|\mathcal{Y}|$*. Define* $g : \mathcal{A}^{\mathcal{Y}} \subset \mathbb{R}^{d|\mathcal{Y}|} \to \mathbb{R}$ *where*

$$g(x_1, x_2, \cdots, x_{|\mathcal{Y}|}) = \mathbb{E}_{y'}[\|x_i - x_{y'}\|^2], y' \in \{1,2,\cdots,|\mathcal{Y}|\} - i \tag{127}$$

is Lipschitz in x, where $\mathcal{A} = \{x \in \mathbb{R}^d : \|x\|_2 \leq F_T B\}$*.*

Proof:

$$\nabla_{x_i} g(x_1, x_2, \cdots, x_{|\mathcal{Y}|}) = 2\mathbb{E}_{y'}[(x_i - x_{y'})] \tag{128}$$
$$\nabla_{x_j} g(x_1, x_2, \cdots, x_{|\mathcal{Y}|}) = 2p(x_j)(x_j - x_i), j \neq i \tag{129}$$

$$\|\nabla_x g(x)\| \leq \left\|\nabla_{x_i} g(x_1, x_2, \cdots, x_{|\mathcal{Y}|})\right\| + \sum_{j \neq i} \left\|\nabla_{x_j} g(x_1, x_2, \cdots, x_{|\mathcal{Y}|})\right\| \tag{130}$$
$$\leq 2\mathbb{E}_{y'}[\|x_i - x_{y'}\|] + 2\sum_{k \neq i} \|x_k - x_i\| \leq 2F_T B|\mathcal{Y}| \tag{131}$$

We know

$$\mathcal{L}^y(\theta_y) = \frac{1}{2}\sum_{j=1}^{N} \frac{\lambda(t_j)(t_j - t_{j-1})}{\bar{\sigma}_{t_j}} \mathbb{E}_{X_0}\mathbb{E}_{\xi_j}\left[\left\|\bar{\sigma}_{t_j} s_{t_j,\theta_y}(x(t_j)) + \xi_j\right\|_2^2\right] \tag{132}$$
$$+ \frac{1}{2}\sum_{j=1}^{N} \lambda(t_j)(t_j - t_{j-1}) C_{t_j}(y) \tag{133}$$

where $C_t(y) = \mathbb{E}_{X_t}\|\nabla \log p_t(.|y)\|^2 - \mathbb{E}_{X_0}\mathbb{E}_{X_t|X_0}\|\nabla \log p_t(x_t|x_0, y)\|^2$. Let $\bar{C}(y) = \frac{1}{2}\sum_{j=1}^{N} \lambda(t_j)(t_j - t_{j-1}) C_{t_j}(y)$

Lemma 9. *With probability $1 - Nn_y\delta$, an upper bound for the generalization gap i.e.*

$$\sup_{(\theta_y, \theta_{-y})} |\ \mathcal{L}^y_{reg}(\theta_y, \theta_{-y}) - \bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y})\ | \tag{134}$$

is

$$\frac{2\sqrt{2}BF_T}{\sqrt{mn_y}} L_{C_1} \sum_{j=1}^{N} \frac{\lambda(t_j)(t_j - t_{j-1})}{\bar{\sigma}_{t_j}} + \frac{2\sqrt{2}BF_T|\mathcal{Y}|^2}{\sqrt{mn_y}} L_{C_2} \sum_{j=1}^{N} \omega(t_j)(t_j - t_{j-1}) + \bar{\mathcal{C}} \tag{135}$$

where $L_{C_1} = \bar{\sigma}^2_{t_j} BF_T + \sqrt{d}\sqrt{\log \frac{2}{\pi\delta^2}})$,$L_{C_2} = 2F_T B|\mathcal{Y}|$,$\bar{\mathcal{C}} = \max_{y \in \mathcal{Y}} |\bar{C}(y)|$

Proof. Observe that, we can rewrite Eq. 134 using triangle inequality as

$$\sup_{(\theta_y, \theta_{-y})} |\ \mathcal{L}^y_{reg}(\theta_y, \theta_{-y}) - \bar{\mathcal{L}}^{n_y}_{reg}(\theta_y, \theta_{-y})\ | \leq \sup_{\theta_y} |\mathcal{L}^y(\theta_y) - \bar{\mathcal{L}}^{n_y}(\theta_y)| + \beta \sup_{(\theta_y, \theta_{-y})} |\mathcal{L}^y_{mut}(\theta_y, \theta_{-y}) - \bar{\mathcal{L}}^{n_y}_{mut}(\theta_y, \theta_{-y})| \tag{136}$$

Further decomposing them, we get

$$\sup_{\theta_y} |\mathcal{L}^y(\theta_y) - \bar{\mathcal{L}}^{n_y}(\theta_y)| \leq \sum_{j=1}^{N} \frac{\lambda(t_j)(t_j - t_{j-1})}{\bar{\sigma}_{t_j}} \sup_{\theta_y} |\mathcal{L}^y(\theta_y)(j) - \bar{\mathcal{L}}^{n_y}(\theta_y)(j)| + \bar{\mathcal{C}} \tag{137}$$

where $|\mathcal{L}^y(\theta_y)(j) - \bar{\mathcal{L}}^{n_y}(\theta_y)(j)| = | \frac{1}{2n_y}\sum_{i=1}^{n_y}\left[\left\|\bar{\sigma}_{t_j} s_{t_j,\theta_y}(x_i(t_j)) + \xi_{ij}\right\|_2^2 - \mathbb{E}_{X_0}\mathbb{E}_{\xi_j}\left\|\bar{\sigma}_{t_j} s_{t_j,\theta_y}(e^{-t_j}X_0 + \sqrt{1 - e^{-2t_j}}\xi_j) + \xi_j\right\|^2\right] |$ and

$$\sup_{(\theta_y, \theta_{-y})} |\mathcal{L}^y_{mut}(\theta_y, \theta_{-y}) - \bar{\mathcal{L}}^{n_y}_{mut}(\theta_y, \theta_{-y})| \leq \sum_{j=1}^{N} \omega(t_j)(t_j - t_{j-1}) \sup_{(\theta_y, \theta_{-y})} |\mathcal{L}^y_{mut}(\theta_y, \theta_{-y})(j) - \bar{\mathcal{L}}^{n_y}_{mut}(\theta_y, \theta_{-y})(j)\ | \tag{138}$$

where

$$|\mathcal{L}^{y}_{mut}(\theta_y,\theta_{-y})(j) - \bar{\mathcal{L}}^{n_y}_{mut}(\theta_y,\theta_{-y})(j)\ |=|\ \frac{1}{2n_y}\sum_{i=1}^{n_y}\Big[\mathbb{E}_{y'\sim Q}\Big[\left\|s_{t_j,\theta_y}(x_i(t_j)) - s_{t,\theta_{y'}}(x_i(t_j))\right\|_2^2 \\ - \mathbb{E}_{X_0}\mathbb{E}_{\xi_j}\mathbb{E}_{y'\sim Q}[\left\|s_{t_j,\theta_y}(e^{-t_j}X_0+\sqrt{1-e^{-2t_j}}\xi_j) - s_{t_j,\theta_{y'}}(e^{-t_j}X_0+\sqrt{1-e^{-2t_j}})\right\|^2]\Big]\ | \tag{139}$$

Finally using Corollary 2, Lemmas 5 7,8, [17, Section 4.1] we have

$$\sup_{\theta_y}|\mathcal{L}^y(\theta_y) - \bar{\mathcal{L}}^{n_y}(\theta_y)| \le \frac{2\sqrt{2}BF_T}{\sqrt{mn_y}}L_{C_1}\sum_{j=1}^{N}\lambda(t_j)(t_j - t_{j-1}) + \bar{\mathcal{C}} \tag{140}$$

and

$$\sup_{(\theta_y,\theta_{-y})}|\mathcal{L}^{y}_{mut}(\theta_y,\theta_{-y}) - \bar{\mathcal{L}}^{n_y}_{mut}(\theta_y,\theta_{-y})| \le \frac{2\sqrt{2}BF_T|\mathcal{Y}|^2}{\sqrt{mn_y}}L_{C_2}\sum_{j=1}^{N}\omega(t_j)(t_j - t_{j-1}) \tag{141}$$

C Numerical Experiments

Computing resources. The numerical experiments were conducted on a MacBook Air (2023) and Gilbreth. Gilbreth has heterogeneous hardware comprising of Nvidia V100, A100, A10, and A30 GPUs in separate sub-clusters. All the nodes are connected by 100 Gbps Infiniband interconnects. We used sub-cluster B with 16 nodes, 24 cores per node, 192 GB memory per node, 3 A30 (24 GB) per node. For more information follow this link.

The width of network $m = 16$, learning rate $\eta_\tau = 10^{-4}, \forall\tau, T_{train} = 5000$ is fixed for Adam optimizer. We set $\lambda(t) = \bar{\sigma}_t, \omega(t) = e^t$, total number of training samples is 50.

Case one We perform more empirical experiments on $d = 1$, imbalance ratio $r = 2.5, \beta = 0.01$. We compute the KL-divergence between the ground truth distribution and the learned model using the procedure in [15]. $P(x|y=1) \sim \mathcal{N}(-\mu,\sigma^2)$ and class 2 is $P(x|y=2) \sim \mathcal{N}(\mu,\sigma^2)$. We observe Fig. 2 the worst case KL divergence for the mutual learning case is lower than the vanilla when we change the distance between mean and the variance of each class label. The performance of head class doesn't worsened for small μ. However, the head class performance suffers for mutual learning case when the distance between the mean increases. This might be because when the support of class distribution are farther apart mutual learning is not advantageous as transfer of knowledge between the class is not useful.

Case two We now consider a case with two classes with imbalance ratio $r = 2.5, \beta = 0.01$. Class 1 itself is a uniform mixture of two Gaussian i.e $P(x|y=1) \sim \frac{1}{2}\mathcal{N}(-4,3) + \frac{1}{2}\mathcal{N}(4,3)$ and class 2 is $P(x|y=2) \sim \mathcal{N}(0,2)$ as in Fig. 3. We observe the Mutual Learning objective with our formulation have lower KL-divergence for both the classes compared to the vanilla diffusion models trained on each class. In this case, mutual learning allows useful transfer of knowledge between the classes increasing the performance for both. We hypothesize that under some notion of similarity between various class distributions, mutual learning is advantageous in improving the performance of all classes.

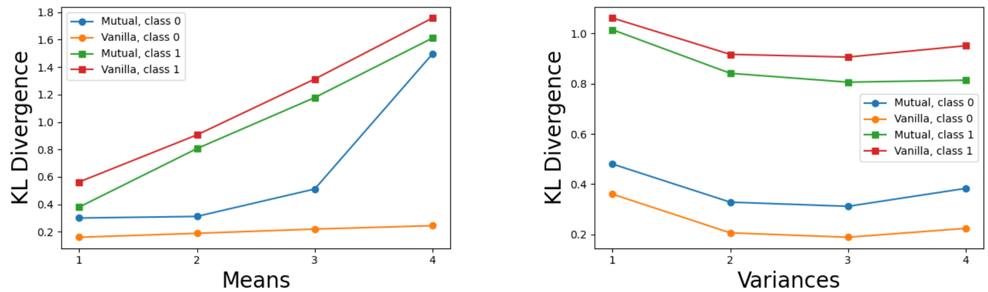

Figure 2: Case one: (Left) The first plot shows the KL-divergence for each class with and without mutual learning objective as μ is varied. (Right) shows the KL-divergence for each class with and without mutual learning objective as σ is varied ($\mu = 2$ fixed).

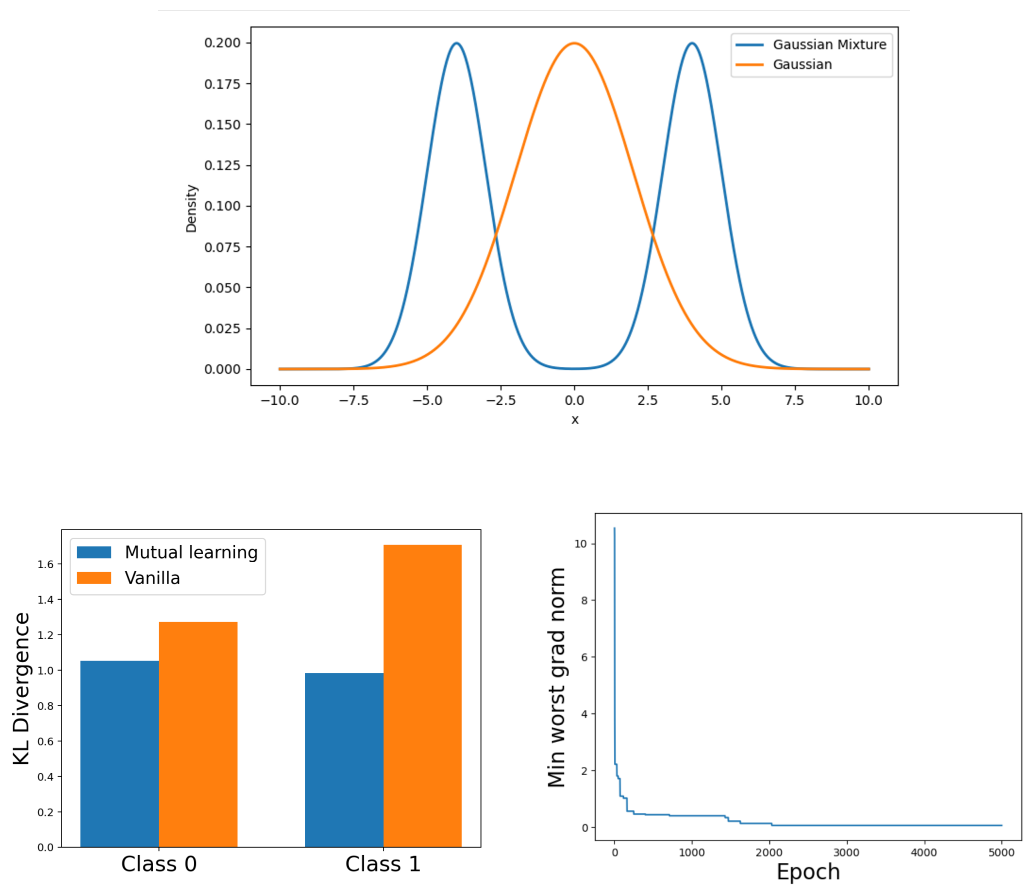

Figure 3: Case two: (Top) The first plot shows class 1 as a gaussian mixture with class 2 as Gaussian. (Bottom Left) Shows the KL-divergence for each class with and without mutual learning objective. (Bottom Right) Shows $\min_{\tau} \max_{y \in \mathcal{Y}} \left\| \nabla \bar{\mathcal{L}}_{reg}^{n_y}(\theta_y^{\tau}, \theta_{-y}^{\tau}) \right\|$ decreasing with training epoch.