Preventing Model Collapse via Contraction-Conditioned Neural Filters

Zongjian Han [*1] **Yiran Liang** [*2] **Yiwei Luo** [*3] **Ruiwen Wang** [*4] **Yilin Huang** [5]

Abstract

This paper presents a neural network filter method based on contraction operators to address model collapse in recursive training of generative models. Unlike (Xu et al., 2024), which requires superlinear sample growth ($O(t^{1+s})$), our approach completely eliminates the dependence on increasing sample sizes within an unbiased estimation framework by designing a neural filter that learns to satisfy contraction conditions. We develop specialized neural network architectures and loss functions that enable the filter to actively learn contraction conditions that satisfy Assumption 2.3 in exponential family distributions, thereby ensuring practical application of our theoretical results. Theoretical analysis demonstrates that when the learned contraction conditions are satisfied, estimation errors converge probabilistically even with constant sample sizes, i.e., $\limsup_{t\to\infty} \mathbb{P}(\|\mathbf{e}_t\| > \delta) = 0$ for any $\delta > 0$. Experimental results show that our neural network filter effectively learns contraction conditions and prevents model collapse under fixed sample size settings, providing an end-to-end solution for practical applications.

1. Introduction

1.1. The Challenge of Model Collapse in the Age of Synthetic Data

The unprecedented scaling of large language models (Achiam et al., 2023; Touvron et al., 2023) has created an insatiable demand for training data, far exceeding the available supply of high-quality, human-generated content (Villalobos et al., 2024). To address this gap, researchers and practitioners increasingly turn to **synthetic data**—data generated by the models themselves. However, this practice poses a critical risk: as synthetic data proliferates online and is inevitably incorporated into future training cycles, it can trigger a degenerative process known as **model collapse** (Shumailov et al., 2024).

Model collapse refers to the progressive degradation in model performance and diversity when generative models are iteratively trained on their own synthetic outputs (Shumailov et al., 2024; Alemohammad et al., 2024). As illustrated in Figure 3, this recursive process causes the model to gradually forget the original data distribution, leading to a final model that generates homogeneous and often meaningless outputs.

1.2. Foundations: Models, Parameters, and Recursive Training

To establish a precise foundation for our work, we first define the key concepts:

- **Model**: In this paper, we consider a **parametric generative model** $\mathbb{P}_{\boldsymbol{\theta}}$, which is a probability distribution over data points $\boldsymbol{x} \in \mathcal{X}$, completely determined by a parameter vector $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. For example, in a Gaussian model, $\boldsymbol{\theta}$ would represent the mean and covariance parameters.
- **Model Parameters**: The vector $\boldsymbol{\theta}$ that defines the model's behavior. The goal of training is to estimate these parameters from data. We denote the **true parameters** that generate real data as $\boldsymbol{\theta}^*$, and the estimated parameters at step t as $\widehat{\boldsymbol{\theta}}_t$.
- **Recursive Training Workflow**: The standard process that leads to model collapse follows this iterative pattern:
  1. Start with real data $\mathcal{D}_0 = \{\boldsymbol{x}_{0,i}\}_{i=1}^n \sim \mathbb{P}_{\boldsymbol{\theta}^*}$
  2. For $t = 1, 2, \ldots, T$:
     - Train model $\mathbb{P}_{\widehat{\boldsymbol{\theta}}_t}$ on dataset $\mathcal{D}_{t-1}$
     - Generate new synthetic data $\mathcal{D}_t = \{\boldsymbol{x}_{t,i}\}_{i=1}^{n_t} \sim \mathbb{P}_{\widehat{\boldsymbol{\theta}}_t}$

[*]Equal contribution [1]School of Mathematical Sciences, Tonji University, Shanghai, 200092, P. R. China [2]School of Mathematical Sciences, Nankai University, Tianjin, China [3]Independent Researcher [4]**AUTHORERR: Missing \icmlaffiliation.** [5]Dundee International Institute, Central South University, Changsha, China. Correspondence to: Firstname1 Lastname1 <first1.last1@xxx.edu>, Firstname2 Lastname2 <first2.last2@www.uk>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

– Use $\mathcal{D}_t$ to estimate parameters for the next generation: $\widehat{\boldsymbol{\theta}}_{t+1} = \mathcal{M}(\mathcal{D}_t)$

This workflow creates a chain: $\mathbb{P}_{\boldsymbol{\theta}^*} \to \mathbb{P}_{\widehat{\boldsymbol{\theta}}_1} \to \mathbb{P}_{\widehat{\boldsymbol{\theta}}_2} \to \cdots \to \mathbb{P}_{\widehat{\boldsymbol{\theta}}_T}$, where each model is trained on data from the previous generation.

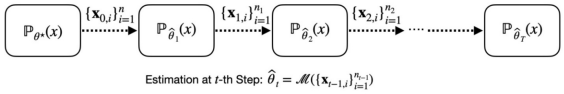

Figure 1. A General Framework for Recursive Training with Fully Synthetic Data.(Xu et al., 2024)

1.3. A Probabilistic Foundation: Random Walks in Parameter Space

To demystify why this recursive workflow leads to collapse, (Xu et al., 2024) provide a powerful probabilistic perspective. They conceptualize recursive training as a **random walk of the model parameters** in the parameter space. This framework elegantly captures the core of the problem:

- **The State**: The estimated parameter $\widehat{\boldsymbol{\theta}}_t$ at generation t is the current position of the walker.
- **The Step**: The transition from $\widehat{\boldsymbol{\theta}}_{t-1}$ to $\widehat{\boldsymbol{\theta}}_t$ is a random step. The direction is determined by the randomness in the finite sample $\mathcal{D}_{t-1}$ drawn from the previous model.
- **The Step Size**: Crucially, the variance of this step—its expected size—is governed by the **sample size** n_t. A larger n_t yields a more precise estimate and a smaller step ($\|\widehat{\boldsymbol{\theta}}_t - \widehat{\boldsymbol{\theta}}_{t-1}\|_2 \propto 1/\sqrt{n_t}$).
- **The Drift**: The estimation procedure $\mathcal{M}$ can introduce bias, acting as a consistent force that "drags" the random walk in a particular direction, thereby accelerating its divergence.

Within this framework, model collapse is the inevitable consequence of this random walk **drifting away from the true parameter** $\boldsymbol{\theta}^*$. (Xu et al., 2024) rigorously show that with a fixed sample size ($n_t = n$), the cumulative error diverges: $\lim_{T\to\infty} \mathbb{E}[(\widehat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^*)^2] \to \infty$, and the model's diversity vanishes with high probability.

1.4. The Prevailing Solution: A Theoretically Sound but Practically Limiting Strategy

The random walk perspective leads to a natural solution: to prevent the walk from straying too far, one must **progressively reduce the step size**. (Xu et al., 2024) derive the precise conditions for this, proving that to prevent collapse:

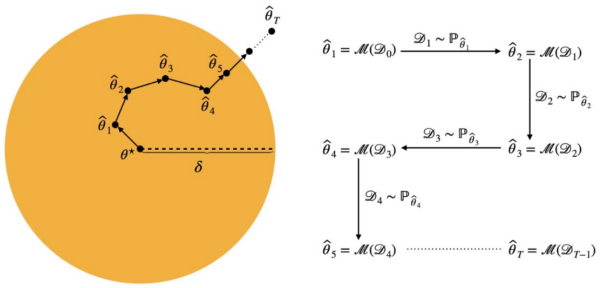

Figure 2. This image explains the principle of model collapse from the perspective of random walks. (Xu et al., 2024)

- For **unbiased estimators**, the sample size must grow at a **superlinear rate**, i.e., $n_t = O(t^{1+s})$ for some $s > 0$.
- For **biased estimators**, an even **faster growth rate** is required, as the bias systematically accelerates the divergence.

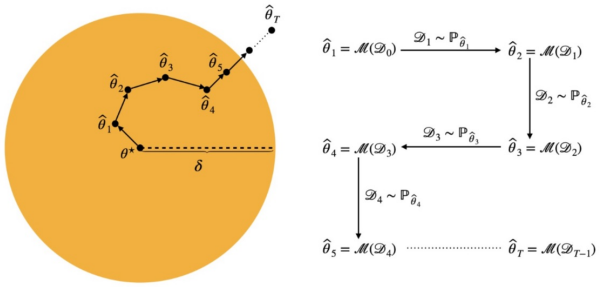

Figure 3. This image explains the principle of model collapse from the perspective of random walks.(Xu et al., 2024)

While this constitutes a critical theoretical milestone, the proposed solution presents a **prohibitive practical barrier**. A superlinear growth schedule implies that computational and storage costs escalate rapidly with each generation. Training over many iterations would become computationally infeasible, rendering this strategy unsuitable for the long-term, sustainable evolution of generative models in real-world scenarios.

1.5. Our Approach: Steering the Random Walk with Learned Contraction Filters

We propose a paradigm shift from *containing* the random walk to *actively steering* it. Instead of using exponentially more data to shrink the step size, we introduce an **neural filter** that learns to correct the walk's direction, ensuring it consistently converges toward the true parameter.

Our Enhanced Workflow: Integrates this filter into the recursive training process:

1. Start with real data $\mathcal{D}_0 \sim \mathbb{P}_{\boldsymbol{\theta}^*}$
2. For $t = 1, 2, \ldots, T$:
   - Generate candidate synthetic data $\mathcal{D}_t^{\text{candidate}} \sim \mathbb{P}_{\widehat{\boldsymbol{\theta}}_t}$
   - Apply neural filter g_ϕ to select a subset $\mathcal{D}_t \subset \mathcal{D}_t^{\text{candidate}}$
   - Estimate next parameters: $\widehat{\boldsymbol{\theta}}_{t+1} = \mathcal{M}(\mathcal{D}_t)$
   - Update filter parameters ϕ to enforce contraction conditions

Our core innovation is to model the error dynamics and enforce a **contraction condition**. We formulate the recursive process as a nonlinear stochastic system:

$$\mathbf{e}_{t+1} = A(\mathbf{e}_t)\mathbf{e}_t + \boldsymbol{\xi}_t', \quad \text{where } A(\mathbf{e}_t) = I - B(\mathbf{e}_t)$$

Here, $\mathbf{e}_t = \widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*$ is the estimation error, and $\boldsymbol{\xi}_t'$ is the estimation noise. The key is to design a neural network that learns a data selection filter such that the state-dependent matrix $A(\mathbf{e}_t)$ satisfies a **contraction condition** (Assumption 2.7), formally expressed as $A(\mathbf{e})^T P A(\mathbf{e}) \preceq (1 - c(\mathbf{e}))P$.

When this condition is met, our theoretical analysis (Theorems 2.7 and 2.12) guarantees that the error process converges in probability: $\limsup_{t\to\infty} \mathbb{P}(\|\mathbf{e}_t\| > \delta) = 0$ for any $\delta > 0$. This holds **even when the sample size n_t is not superlinear**, effectively breaking the superlinear growth requirement established by (Xu et al., 2024).

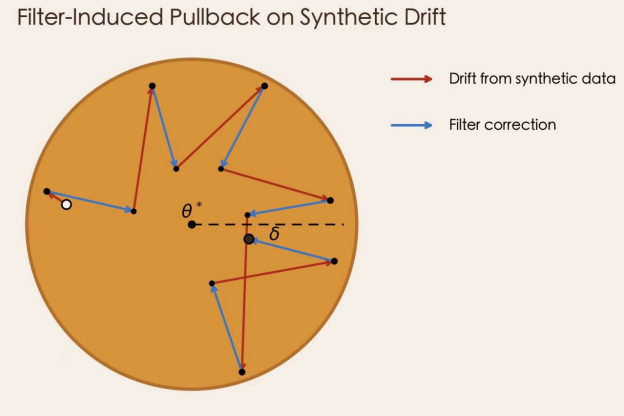

Figure 4. This figure explains the working mechanism of the filter: by continuously pulling the parameters back to the origin, it prevents the model from collapsing.

As illustrated in Figure 4, our filter acts as a guidance system, continually pulling the wandering parameter estimate back towards the basin of attraction around $\boldsymbol{\theta}^*$.

Methodology: After we establish the mathematical theory under some condition, we give a method to design a neural network as the filter which can force the filter satisfies the condition. We first label the data in the first or the initial few steps by human or by machine, then we use these data to train the filter which make the filter force to the condition under the loss function constraint.

1.6. Summary of Contributions

Our work bridges the gap between the theoretical understanding of model collapse and a practical, deployable solution. The main contributions are:

- **A New Theoretical Framework:** We introduce a contraction-operator-based theory that provides a sufficient condition for preventing model collapse with **constant sample sizes**, moving beyond the superlinear growth requirement of (Xu et al., 2024).
- **An End-to-End Algorithmic Solution:** We design a specialized neural network architecture that acts as a data filter. This network is trained with a novel combined loss function to actively **learn the contraction conditions** required by our theory, ensuring theoretical guarantees are met in practice.
- **Integration of Theory and Practice:** We provide a complete framework that translates theoretical stability conditions into a learnable objective for a neural network, ensuring our method is both principled and practical.
- **Empirical Validation:** We demonstrate through experiments that our neural filter successfully prevents model collapse in exponential family distributions under fixed sample size settings, validating its effectiveness and superiority.

2. Mathematical Modeling and Deduction

2.1. Mathematical Modeling

When the estimation is unbiased, we aim to train a filter to collect data to avoid model collapse, which has a function to pull the point back to the origin, which is abstracted as the map B.

Consider the nonlinear stochastic difference system:

$$\mathbf{e}_{t+1} = A(\mathbf{e}_t)\mathbf{e}_t + \boldsymbol{\xi}_t', \quad \text{where } A(\mathbf{e}_t) = I - B(\mathbf{e}_t) \quad (1)$$

and $B(\mathbf{e}_t)$ is a state-dependent contraction operator.

We make the following assumptions to characterize the role of filters in stochastic systems with model collapse. **1.The random walk come form the error generated by estimator:**

Assumption 2.1. (Noise Properties):
The noise process $\{\boldsymbol{\xi}_t'\}$ satisfies

1. $\mathbb{E}[\boldsymbol{\xi}_t'|\mathcal{F}_t] = 0$, where $\mathcal{F}_t = \sigma(\mathbf{e}_0, \boldsymbol{\xi}_0', \ldots, \boldsymbol{\xi}_{t-1}')$ is the filtration.
2. there exists a sequence $\sigma_t^2 > 0$ and $\lim_{t\to\infty} \sigma_t^2 = 0$ such that $\mathbb{E}[\boldsymbol{\xi}_t'^T P \boldsymbol{\xi}_t'|\mathcal{F}_t] \leq \sigma_t^2$ almost surely.

Remark 2.2. Actually this is reasonable, by (Xu et al., 2024), when the sample growth (here the growth can be in any speed), the upper bound of the step size in the random walk of parameter will turn to zero.

2. We use a functional c to control the map A:

Assumption 2.3. (contraction condition)

There exists a symmetric positive definite matrix $P \succ 0$ and a continuous function $c : \mathbb{R}^p \to [0, 1)$ such that:

$$A(\mathbf{e})^T P A(\mathbf{e}) \preceq (1 - c(\mathbf{e}))P, \quad \forall \mathbf{e} \in \mathbb{R}^p$$

Remark 2.4. The function C can be seen as the shrinking strength, where the strength increases with the increase of error seen in figure 5.

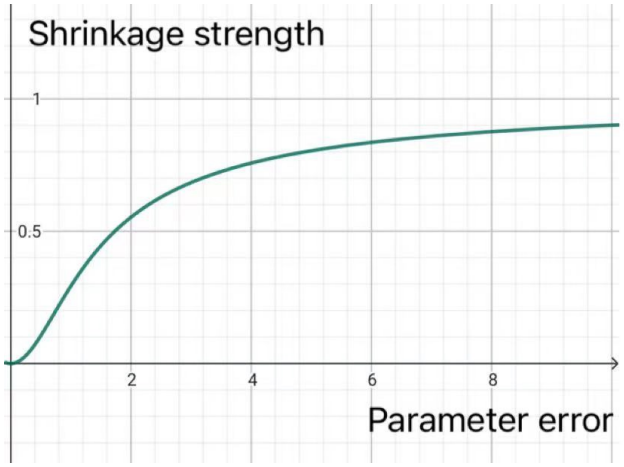

Figure 5. This figure show the relationship between the contraction strength and the vector distance from the origin which is the parameter error.

3. We introduce a convex function f to regulate the contraction function c, ensuring that the farther away from the origin, the stronger the pulling back effect:

Assumption 2.5. (Properties of the Contraction Function):

There exists a convex function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ with $f(0) = 0$, $f(r) > 0$ for $r > 0$, and

$$c(\mathbf{e})V(\mathbf{e}) \geq f(V(\mathbf{e})),$$

here $V(\mathbf{e}) := \mathbf{e}^T P \mathbf{e}$ is the Lyapunov function.

The function c and f are exist,

Example: We can just take $c : \mathbf{e} \mapsto 1 - (V(\mathbf{e}) + 1)^{-\frac{1}{2}}$ and $f : r \mapsto r(1 - (r + 1)^{-\frac{1}{2}})$ can satisfies the condition.

Remark 2.6. Here the function f is used to regular the function C, and we need not to worry about the wether the condition can be satisfied, we will construct a loss function to make the filter converge the condition in section 3.2.

2.2. Main Theoretical Results

Here we prove that this random system with a pullback effect, when the square of the norm of the step length of the random walk is controlled by a sequence that converges to 0, will eventually enter any neighborhood of the origin.

Theorem 2.7. *If a stochastic difference system 1 satisfies Assumption 2.3, 2.1 and 2.5, then for each $\delta > 0$ $\limsup_{t\to\infty} \mathbb{P}(\|\mathbf{e}_t\| > \delta) = 0$,* **which means the model will not collapse**.

2.3. The Estimation of convergence rate

We make the following assumptions on the growth of f and the convergence rate of noise, in order to provide the convergence rate of this stochastic system with a pullback effect.

Assumption 2.8 (Function Properties)**.** The function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ satisfies: There exist constants $c_1, c_2 > 0$ and $x_0 > 0$ such that for all $0 < x \leq x_0$:

$$c_1 x^p \leq f(x) \leq c_2 x^p.$$

Assumption 2.9 (Noise Properties)**.** The noise sequence $\{\sigma_t^2\}$ satisfies:

$$\sigma_t^2 = O(t^{-\beta})$$

for some $\beta > 0$.

Theorem 2.10 (Convergence Rates for Recurrence Inequality)**.** *Under Assumptions 2.8 and 2.9, the sequence $\{x_t\}$ satisfies the following convergence rates:*

1. If $p = 1$, then $x_t = O\left(\max\left(e^{-ct}, t^{-\beta}\right)\right)$ for some $c > 0$, where $c = -\log(1 - c_1) > 0$.

2. If $p > 1$, then $x_t = O\left(\max\left(t^{-\frac{1}{p-1}}, t^{-\frac{\beta}{p}}\right)\right)$

2.4. The combination with the work of predecessors

Here we use the assumption and consequence from (Xu et al., 2024) to prove that when the filter exists, sample size n_t need not grow superlinearly.

Assumption 2.11. (Xu et al., 2024) For a class of parametric generative models $\mathcal{P} = \{\mathbb{P}_\theta : \theta \in \Theta\}$, let $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^t$

be a dataset generated from $\mathbb{P}_\theta$ for some $\theta \in \Theta$. Suppose that $\widehat{\boldsymbol{\theta}} = \mathcal{M}(\mathcal{D})$ is an estimate of $\boldsymbol{\theta}$ obtained under the estimation scheme $\mathcal{M}$. Assume there exist constants $C_1, C_2, \gamma > 0$ and a positive diverging sequence $r(t)$ such that for all $t \geq 1$ and any $\delta > 0$, we have

$$\sup_{\theta \in \Theta} \mathbb{P}\left(\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|_2 \geq \delta \right) \leq C_1 \exp(-C_2 r(t) \delta^\gamma),$$

where the probability is taken over the randomness of $\mathcal{D}$ generated i.i.d. from $\mathbb{P}_\theta(\boldsymbol{x})$.

Theorem 2.12. *Suppose that $\widehat{\boldsymbol{\theta}} = \mathcal{M}(\mathcal{D})$ is an estimate of $\boldsymbol{\theta}$ with $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^t \sim \mathbb{P}_{\boldsymbol{\theta}}$ satisfying Assumption 2.11 with $r(t) = t^\kappa$, where $\kappa > 0$. Then for each $\delta > 0$,*

$$\limsup_{t \to \infty} \mathbb{P}(\|\mathbf{e}_t\| > \delta) = 0,$$

which means the model will not collapse.

Remark 2.13. Theorem 2.12 tell us that when we have a filter which satisfies Assumption 2.3 and 2.5, then we only need a lower growth speed thar can make sure the model would not collapse.

3. Neural Architecture Design

We propose a neural network-based filter designed to select data points for parameter estimation in exponential family distributions. When we constructed c, f, P in Assumption 2.3 and 2.5, the filter is trained via gradient descent to satisfy contraction condition (Assumption 2.3) on the estimation error while maintaining classification accuracy against human labels. Our solution includes the network architecture, loss functions, and training procedure..

Let $\mathbb{P}_\theta$ be a distribution and we have samples $\{\boldsymbol{x}_{\text{est},i}\}_{i=1}^n$ and a unbiased estiamtion $\mathcal{M}$. then we define $\hat{\theta}_{\text{est}} := \mathcal{M}(\{\boldsymbol{x}_{\text{est},i}\}_{i=1}^n)$.

For axemplr for exponential family distributions

$$f(x|\theta) = h(x) \exp\left(\theta^T T(x) - \Phi(\theta)\right)$$

we can use

$$\hat{\theta}_{\text{est}} = (\nabla \Phi)^{-1}(\bar{T}_{\text{est}})$$

to approximate its parameters, where

$$\bar{T}_{\text{est}} = \frac{1}{n} \sum_{i=1}^n T(x_i)$$

is the sufficient statistic for the sample.

3.1. Problem Statement

Given a dataset $D = \{x_i\}_{i=1}^N$ from an distribution with unknown true parameter θ_{true}, we design a filter $g_\phi(\cdot)$ that:

1. Outputs selection probabilities $w_i \in [0, 1]$ for each data point based on (PCA-reduced features z_i, if we use PCA) data component.
2. Satisfies the contraction condition:

$$\mathbf{e}_{\text{new}}^\top P \mathbf{e}_{\text{new}} \leq (1 - c(\mathbf{e}_{\text{est}})) \mathbf{e}_{\text{est}}^\top P \mathbf{e}_{\text{est}} \tag{2}$$

   where $\mathbf{e}_{\text{est}} = \theta_{\text{est}} - \theta_{\text{good}}$, $\mathbf{e}_{\text{new}} = \theta_{\text{new}} - \theta_{\text{good}}$, $P \succ 0$ is a symmetric positive definite matrix, and $c(\mathbf{e})$ is a continuous function.
3. Minimizes classification error against human-provided labels $y_i \in \{0, 1\}$ (0: bad sample, 1: good sample).

Key Components and Assumptions

1. Data Preparation
   - Dataset: $D = \{x_i\}_{i=1}^N$ with labels y_i which are labeled as good sample. (The label can be formed from human or the machine. In our experiment we use machine to label sample.)
   - If the dimension of data is too large, we can use PCA to raw features to obtain reduced-dimensional features $z_i \in \mathbb{R}^d$.
2. Parameter Estimation
   - **Initial estimate** θ_{est}:

$$\theta_{\text{est}} = \mathcal{M}(\{\boldsymbol{x}_{\text{est},i}\}_{i=1}^n)$$

   - **Reference estimate** θ_{good} (using good samples only):

$$\theta_{\text{good}} = \mathcal{M}(\{\boldsymbol{x}_{\text{good},i}\}_{i=1}^m)$$

   - **Error vectors**: $\mathbf{e}_{\text{est}} = \theta_{\text{est}} - \theta_{\text{good}}$, $\mathbf{e}_{\text{new}} = \theta_{\text{new}} - \theta_{\text{good}}$
3. Filter Operation The filter defines a *pullback operator* B implicitly through data selection:

$$A = I - B, \quad \mathbf{e}_{\text{new}} = A \mathbf{e}_{\text{est}}$$

   where A is the matrix satisfying the contraction condition (2).

3.2. Neural Architecture Design

- **Architecture**: Multi-layer Perceptron (MLP)
- **Input**: Data or PCA-reduced features $z_i \in \mathbb{R}^d$
- **Hidden layers**: ≥ 1 layer with ReLU activation (e.g., dimension 128)
- **Output**: Single node with sigmoid activation $\rightarrow w_i = g_\phi(z_i)$

- **Parameters**: Weights and biases denoted as ϕ

Loss Function Total loss combines classification and contraction objectives:

$$L_{\text{total}}(\phi) = L_{\text{class}}(\phi) + \lambda L_{\text{contract}}(\phi) \tag{3}$$

where $\lambda > 0$ is a hyperparameter.

1. Classification Loss Binary cross-entropy with human labels:

$$L_{\text{class}}(\phi) = -\frac{1}{N}\sum_{i=1}^{N}\Big[y_i \log(g_\phi(z_i)) + (1-y_i)\log(1-g_\phi(z_i))\Big] \tag{4}$$

2. Contraction Loss Hinge loss enforcing (2):

$$L_{\text{contract}}(\phi) = \max\Big(0,\ \mathbf{e}_{\text{new}}^\top P \mathbf{e}_{\text{new}} - (1 - c(\mathbf{e}_{\text{est}}))\mathbf{e}_{\text{est}}^\top P \mathbf{e}_{\text{est}}\Big) \tag{5}$$

where:

- $\mathbf{e}_{\text{est}}$ is fixed during training
- $\mathbf{e}_{\text{new}}$ depends on ϕ via θ_{new}
- $c(\mathbf{e})$ is predefined (e.g., $c(\mathbf{e}) = \alpha\|e\|_2^2$)

3.3. Training Procedure

We train the network as follows, which is flexible and depend on the design of researcher. Here we use the exponential family distributions for example.

Algorithm 1 Filter Training

1: **Initialize:**
- Compute θ_{est}, θ_{good} from D
- Extract PCA features z_i for all x_i
- Initialize filter parameters ϕ

2: **for** epoch = 1 to MaxEpochs **do**

3: **Forward Pass:**
- Compute weights: $w_i = g_\phi(z_i)$
- Compute weighted sufficient statistic:

$$\bar{T}_{\text{new}} = \frac{\sum_{i=1}^{N} w_i T(x_i)}{\sum_{i=1}^{N} w_i}$$

- Solve for θ_{new}:

$$\theta_{\text{new}} = (\nabla\Phi)^{-1}(\bar{T}_{\text{new}}) \quad \text{(analytic/numerical)}$$

- Compute error: $\mathbf{e}_{\text{new}} = \theta_{\text{new}} - \theta_{\text{good}}$

4: **Loss Calculation:**

$$L_{\text{total}} = L_{\text{class}} + \lambda L_{\text{contract}}$$

5: **Backward Pass:**

$$\phi \leftarrow \phi - \eta \nabla_\phi L_{\text{total}} \quad \text{(via Adam optimizer)}$$

6: **end for**

3.4. Convergence Analysis

Existence of Solution Assume:

- $\exists \phi^*$ such that $L_{\text{class}}(\phi^*) = 0$ and $L_{\text{contract}}(\phi^*) = 0$
- Neural network has sufficient capacity (Universal Approximation Theorem)

Then gradient descent may converge to ϕ^* satisfying the contraction condition.

Optimization Convergence

- $L_{\text{total}} \geq 0$ but non-convex $\rightarrow$ converges to local minimum
- L_{contract} provides gradient when contraction is violated:

$$\nabla_\phi L_{\text{contract}} \propto \nabla_\phi \left(\mathbf{e}_{\text{new}}^\top P \mathbf{e}_{\text{new}}\right) \quad \text{if } L_{\text{contract}} > 0$$

- Monitor $L_{\text{contract}} \rightarrow 0$ during training

4. Experiments

We design experiments to show the effect of filter, we use a neural network to collect data so that we can have enough sample to train the filter. we will compare the offset of parameters between whit and with out filter in three kind of sample growth rate.

4.1. Experimental Setup

Now we have a set of initial data generated by a normal distribution.

We first simulate the workflow for several step and we labeled the 70% of points closest to the true parameters as 'good' and the remainder are marked 'bad' . Then we take these data to training the filter. finally we add the filter into the workflow and recode the **expected distance $\mathbb{E}[(\widehat{\boldsymbol{\theta}}_T - \boldsymbol{\theta}^*)^2]$, between the parameters and the true parameters** when the sample grows.

4.2. Training data

The training data used in this study was entirely generated by simulation, aiming to reproduce the sample quality degradation caused by distribution drift during the iterative parameter estimation process. We construct datasets respectively with four dimension configurations (2 and 3 dimensions), and the data of each dimension is based on the same set of default hyperparameters: The true mean vector is initialized to a constant of 1 ($\mu_0 = 1.0$), the covariance is determined by the identity matrix (standard deviation $\sigma_0 = 1.0$), the number of iteration rounds n_{rounds}, the base sample size of each round n_{samples} , and the contamination rate shows that the filter effectively guides the generated data toward the true distribution. In the specific process, in the t-round, 1000 candidate points will be sampled around the current estimate, sorted in ascending order by the Euclidean distance from the true mean, and the closest 1- contamination rate will be marked as the "good sample" (label 1). The rest contamination rate is marked as "bad sample" (label 0). Subsequently, only by making good use of the mean update of the samples θ_{t+1}, systematic offsets are injected round by round and cumulative drifts in real scenarios are created.

4.3. Experimental Process

After obtaining the data, the merged dataset is standardized and PCA dimensionality reduction is performed. Then, a two-layer MLP (FilterNet) is used to jointly optimize with binary cross-entropy, contraction constraints, and ESS regularization to learn to distinguish between samples that are "helpful for estimation" and those that "cause offset". After the training was completed, we divided the iterative estimation process under the same initial conditions into two groups of control experiments: "no filter" and "loaded filter". The former directly updated the parameters by averaging all samples, while the latter first weighted the samples through the trained filter and then updated them.

4.4. Results and Analysis

Figure 6 shows that the filter will pull the generated data to the real case effectively. And we can see that the parameters tend to converge to the true parameters.

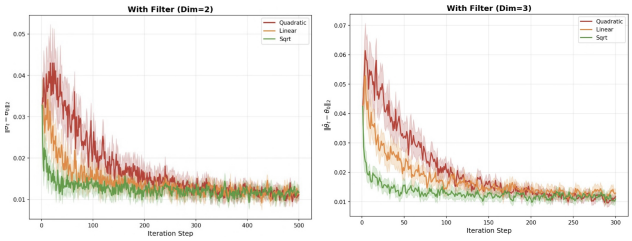

Figure 6. Expectation distance between the parameters and the true parameters when the sample grows in 2 dimension normal distribution

4.5. Ablation Studies

we compare the **eliminated the dependence on superlinear sample growth and initial parameters** when the sample grows at two different rates, **with and without filters**.

Figure 7 shows that when the sample grows at a square root rate, if there is no filter the parameter will run to the infinity away from the real parameter which is corresponding to the theory in (Xu et al., 2024) and if there is a filter then the parameter will converge to the real parameter.

Figure 8 shows that when the sample grows at a quadratic rate, if there is no filter the parameter will converge to a constant distant away from the real parameter which is also corresponding to the theory in (Xu et al., 2024) and if there is a filter then the parameter will converge to the real parameter.

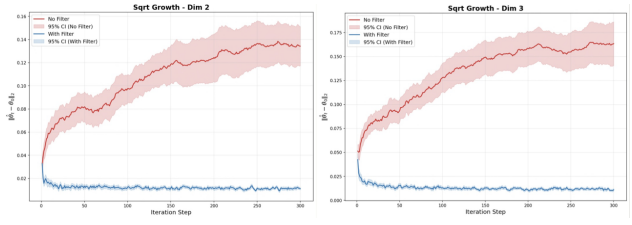

Figure 7. Expectation distance between the parameters and the true parameters when the sample grows in $t^{\frac{1}{2}}$ speed with and without filters in 2 and 3 dimension normal distribution

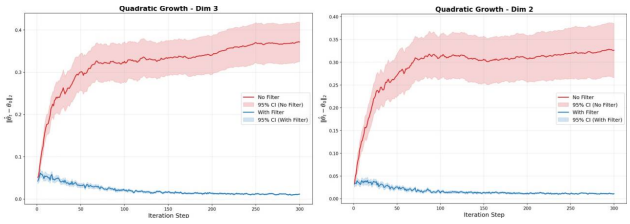

Figure 8. Expectation distance between the parameters and the true parameters when the sample grows in t^2 speed with and without filters in 2 and 3 dimension normal distribution

5. Discussion

5.1. Theoretical Implications

Our work establishes a novel theoretical framework for preventing model collapse in recursive training of generative models, grounded in contraction operators and Lyapunov stability theory. By formulating the error dynamics as a nonlinear stochastic system and enforcing a contraction condition through a learned neural filter, we demonstrate that model collapse can be avoided even with arbitrary sample growth and arbitrary initial sample. This represents a significant departure from the superlinear sample growth requirement established by (Xu et al., 2024), offering a more practical and scalable solution for long-term model training.

Our theoretical results (Theorems 2.7 and 2.12) show that the convergence in probability of the estimation error can be achieved under mild assumptions on the noise (which is depend on the work of (Xu et al., 2024)) and contraction function. This implies that the filter can actively steer the parameter estimate back toward the true parameter, effectively acting as a stabilizing controller in the recursive training loop.

To ensure that the mathematical theory works, we have designed a loss function that enables the filter to converge to the hypothesis. After the data labeled under the simulated working flow was used as the filter for training, an excellent screening effect was achieved, effectively preventing the model from collapsing. When the sample is growing superlinearly, although (Xu et al., 2024) can ensure that the sample parameters do not deviate too far with a sufficient sample size, the requirement for the initial sample size is directly abandoned after adding the filter, and there is an extremely significant improvement compared to not adding the filter.

5.2. Practical Applications

The proposed neural filter is particularly relevant in scenarios where high-quality human-generated data is scarce or expensive to obtain. For example:

1. **Large Language Models (LLMs):**

   As models like GPT-4 and beyond continue to scale, synthetic data will play an increasingly critical role. Our filter can help maintain model quality and diversity over multiple generations of self-training.

2. **Data Augmentation and Privacy-Preserving Synthesis:**

   In domains such as healthcare or finance, where real data is sensitive, synthetic data generated under our framework can be used safely without risking model collapse.

3. **Continual Learning:**

   Our approach can be integrated into continual learning pipelines where models are periodically updated with new synthetic data, ensuring long-term stability.

5.3. Limitations

While our method offers a promising solution, several limitations remain:

1. We need to know the specific distribution type and the non-specific estimation method of its parameters. We have not yet established a theory for partial estimation.

2. The filter's performance depends on the availability of a high-quality reference parameter θ_{good}, which may not always be accessible in fully unsupervised settings.

6. Conclusion

6.1. summary

We conducted mathematical modeling on the filter and identified the sufficient conditions for the convergence of its control system. We also designed a neural network for the filter and incorporated the filter into the neural network. This modification performed very well in the experiment in reducing model collapse and broke through the sample's dependence on superlinear growth and initial parameters.

6.2. Future work

In the future, we will consider the theoretical framework when there is partial estimation, and also design neural networks to converge to the theoretical assumptions.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., et al. GPT-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

Alemohammad, S., Casco-Rodriguez, J., Luzi, L., Humayun, A. I., Babaei, H., et al. Self-consuming generative models go MAD. In *The Twelfth International Conference on Learning Representations*, Vienna, Austria, 2024.

Shumailov, I., Shumaylov, Z., Zhao, Y., Papernot, N., Anderson, R., and Gal, Y. AI models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., et al. LLaMA: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023.

Villalobos, P., Ho, A., Sevilla, J., Besiroglu, T., Heim, L., and Hobbhahn, M. Position: Will we run out of data? limits of LLM scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*, pp. 1–15, 2024.

Xu, S., He, H., and Cheng, G. A probabilistic perspective on model collapse. arXiv preprint arXiv:2505.13947, 2024.

A. Appendix

A.1. proof of Theorem 2.7

Notice that $\lim_{t\to\infty} \sigma_t^2 = 0$ which means for each $\varepsilon > 0$, there exist N for each $t > N$, we have

$$\sigma_t^2 < \varepsilon.$$

Now we fix ε and begin from $t = N + 1$.

Taking unconditional expectation in Assumption 1:

$$\mathbb{E}[V(\mathbf{e}_{t+1})] \leq \mathbb{E}[V(\mathbf{e}_t)] - \mathbb{E}[c(\mathbf{e}_t)V(\mathbf{e}_t)] + \varepsilon.$$

By Assumption 3, $c(\mathbf{e}_t)V(\mathbf{e}_t) \geq f(V(\mathbf{e}_t))$, so:

$$\mathbb{E}[V(\mathbf{e}_{t+1})] \leq \mathbb{E}[V(\mathbf{e}_t)] - \mathbb{E}[f(V(\mathbf{e}_t))] + \varepsilon.$$

Notice that f is a convex function, then we have

$$\mathbb{E}[V(\mathbf{e}_{t+1})] \leq \mathbb{E}[V(\mathbf{e}_t)] - f(\mathbb{E}[V(\mathbf{e}_t)]) + \varepsilon.$$

Let $x_t := \mathbb{E}[V(\mathbf{e}_{t+1})]$ and $b := \varepsilon$ then we get

$$x_{t+1} \leq x_t - f(x_{t+1}) + b.$$

then we have

Lemma A.1. *Consider a nonnegative sequence $\{x_t\}$ satisfying the recurrence inequality:*

$$x_{t+1} \leq x_t - f(x_t) + b,$$

where $b > 0$, and f is a positive convex function (i.e., convex and $f(x) > 0$ for all $x > 0$). Assume that f is continuous, and the equation $f(x) = b$ has a largest solution L (i.e., $L = \max\{x \geq 0 : f(x) = b\}$). Then the limit superior of the sequence satisfies:

$$\limsup_{t\to\infty} x_t \leq L.$$

Proof. Assume, for contradiction, that $\limsup_{t\to\infty} x_t = U > L$. By the definition of limit superior, there exists a subsequence $\{x_{t_k}\}$ such that:

$$\lim_{k\to\infty} x_{t_k} = U.$$

Since $U > L$, choose $\epsilon > 0$ such that $U > L + \epsilon$. Then for sufficiently large k, we have:

$$x_{t_k} > U - \frac{\epsilon}{2} > L + \frac{\epsilon}{2}.$$

Because f is continuous and L is the largest solution of $f(x) = b$, for any $x > L$, we have $f(x) > b$. In particular, for $x > L + \epsilon/2$, $f(x) > b$. Since $x_{t_k} \to U$ and f is continuous, it follows that:

$$\lim_{k\to\infty} f(x_{t_k}) = f(U) > f(L) = b.$$

Thus, there exists $\delta > 0$ and $K_1 \in \approx$ such that for all $k \geq K_1$:

$$f(x_{t_k}) \geq b + \delta.$$

Substituting into the recurrence inequality:

$$x_{t_k+1} \leq x_{t_k} - f(x_{t_k}) + b \leq x_{t_k} - (b + \delta) + b = x_{t_k} - \delta.$$

That is:

$$x_{t_k+1} \leq x_{t_k} - \delta.$$

On the other hand, since $x_{t_k} \to U$, for the given $\delta > 0$, there exists $K_2 \in \approx$ such that for all $k \geq K_2$:

$$x_{t_k} < U + \frac{\delta}{2}.$$

Take $k \geq \max\{K_1, K_2\}$. Then:

$$x_{t_k+1} \leq x_{t_k} - \delta < U + \frac{\delta}{2} - \delta = U - \frac{\delta}{2}.$$

However, $\{x_{t_k+1}\}$ is also a subsequence of $\{x_t\}$, so its limit superior must be at least U. This contradicts the fact that $x_{t_k+1} < U - \delta/2$ for all sufficiently large k. Therefore, the initial assumption is false, and we conclude:

$$\limsup_{t\to\infty} x_t \leq L.$$

□

Notice that L is only dependent on b, which is independent on x_0

note that:

$$\{\|\mathbf{e}_t\| > R\} = \{V(\mathbf{e}_t) > \lambda_{\min}(P)R^2\}$$

where $\lambda_{\min}(P) > 0$ is the minimum eigenvalue of matrix P.

This is because:

$$V(\mathbf{e}) = \mathbf{e}^T P\mathbf{e} \geq \lambda_{\min}(P)\|\mathbf{e}\|^2$$

Applying Markov's inequality:

$$\mathbb{P}(\|\mathbf{e}_t\| > R) = \mathbb{P}(V(\mathbf{e}_t) > \lambda_{\min}(P)R^2) \leq \frac{\mathbb{E}[V(\mathbf{e}_t)]}{\lambda_{\min}(P)R^2}$$

Notice thar f is monotonic.

Lemma A.2. *Let $f : [0,\infty) \to [0,\infty)$ be a convex function with $f(0) = 0$. Then f is monotonic. Specifically, f is non-decreasing.*

Proof. Let $0 \leq x < y$. Since f is convex and $f(0) = 0$, we can write x as a convex combination of 0 and y. Let $\lambda = \frac{x}{y} \in [0, 1)$. Then:

$$x = \lambda y + (1 - \lambda) \cdot 0$$

By convexity of f:

$$f(x) = f(\lambda y + (1 - \lambda) \cdot 0) \leq \lambda f(y) + (1 - \lambda)f(0) = \lambda f(y) = \frac{x}{y} f(y)$$

Thus, $f(x) \leq \frac{x}{y} f(y)$.

Now consider two cases:

1. If $f(y) > 0$, then $f(x) \leq \frac{x}{y} f(y) < f(y)$ since $\frac{x}{y} < 1$.
2. If $f(y) = 0$, then $f(x) \leq 0$, but since $f(x) \geq 0$ by the codomain, we have $f(x) = 0 = f(y)$.

In both cases, $f(x) \leq f(y)$ for all $0 \leq x < y$, so f is non-decreasing.

Even if we consider the possibility of f being decreasing, for any $x > 0$ we would have $f(x) \leq f(0) = 0$, and since $f(x) \geq 0$, this implies $f(x) = 0$ for all x, which is constant (and therefore also monotonic).

Hence, in all cases, f is monotonic. □

then we have

$$\limsup_{t\to\infty} \mathbb{P}(\|\mathbf{e}_t\| > R) \leq \frac{f^{-1}(\varepsilon)}{\lambda_{\min}(P)R^2}$$

the right hand can be arbitrary small, thus

$$\limsup_{t\to\infty} \mathbb{P}(\|\mathbf{e}_t\| > R) = 0$$

A.2. proof of Theorem 2.10

Lemma A.3 (Linear Case Comparison)**.** *Consider the recurrence:*

$$y_{t+1} \leq (1-\alpha)y_t + b_t$$

where $0 < \alpha < 1$ *and* $b_t = O(t^{-\beta})$*. Then:*

$$y_t = O\left(\max\left((1-\alpha)^t, t^{-\beta}\right)\right)$$

Proof. The solution can be written as:

$$y_t \leq (1-\alpha)^t y_0 + \sum_{k=0}^{t-1}(1-\alpha)^{t-1-k}b_k$$

Since $b_k \leq Bk^{-\beta} \leq B$ for some constant $B > 0$, the second term is bounded by:

$$\sum_{k=0}^{t-1}(1-\alpha)^{t-1-k}b_k \leq B\sum_{j=0}^{t-1}(1-\alpha)^j \leq \frac{B}{\alpha}$$

For the precise decay rate, we analyze the convolution sum more carefully.

Let $S_t = \sum_{k=0}^{t-1}(1-\alpha)^{t-1-k}b_k$. Since $b_k = O(k^{-\beta})$, there exists $C > 0$ such that:

$$S_t \leq C\sum_{k=1}^{t-1}(1-\alpha)^{t-1-k}k^{-\beta}$$

We split the sum at $k = t/2$:

$$S_t \leq C\left(\sum_{k=1}^{\lfloor t/2\rfloor}(1-\alpha)^{t-1-k}k^{-\beta} + \sum_{k=\lfloor t/2\rfloor+1}^{t-1}(1-\alpha)^{t-1-k}k^{-\beta}\right)$$

For the first sum, since $k \leq t/2$, we have $t-1-k \geq t/2-1$, so:

$$\sum_{k=1}^{\lfloor t/2\rfloor}(1-\alpha)^{t-1-k}k^{-\beta} \leq (1-\alpha)^{t/2-1}\sum_{k=1}^{\infty}k^{-\beta} = O((1-\alpha)^{t/2})$$

For the second sum, since $k \geq t/2$, we have $k^{-\beta} = O(t^{-\beta})$, so:

$$\sum_{k=\lfloor t/2\rfloor+1}^{t-1}(1-\alpha)^{t-1-k}k^{-\beta} = O(t^{-\beta})\sum_{j=0}^{t-1}(1-\alpha)^j = O(t^{-\beta})$$

Therefore, $S_t = O\left(\max\left((1-\alpha)^{t/2}, t^{-\beta}\right)\right)$, and consequently:

$$y_t = O\left(\max\left((1-\alpha)^t, t^{-\beta}\right)\right)$$

□

Lemma A.4 (Polynomial Case Comparison)**.** *Consider the recurrence:*

$$y_{t+1} \leq y_t - cy_t^p + b_t$$

where $p > 1$, $c > 0$, and $b_t = O(t^{-\beta})$. Then:

$$y_t = O\left(\max\left(t^{-\frac{1}{p-1}}, t^{-\frac{\beta}{p}}\right)\right)$$

Proof. We use the comparison function method. Let $\gamma = \min\left(\frac{1}{p-1}, \frac{\beta}{p}\right)$ and define $g(t) = At^{-\gamma}$ for some constant $A > 0$ to be determined.

We want to show by induction that there exists N_0 such that for all $t \geq N_0$, $y_t \leq g(t)$.

Base case: Choose N_0 large enough so that $g(N_0) \leq x_0$ (to ensure the asymptotic bound $f(x) \geq c_1 x^p$ applies) and choose A such that $y_{N_0} \leq g(N_0)$.

Inductive step: Assume $y_t \leq g(t)$ and prove $y_{t+1} \leq g(t+1)$.

From the recurrence:

$$y_{t+1} \leq g(t) - c[g(t)]^p + b_t = At^{-\gamma} - cA^p t^{-\gamma p} + Bt^{-\beta}$$

where $b_t \leq Bt^{-\beta}$ for some $B > 0$.

We need to show:

$$At^{-\gamma} - cA^p t^{-\gamma p} + Bt^{-\beta} \leq A(t+1)^{-\gamma}$$

Using the inequality $(t+1)^{-\gamma} \geq t^{-\gamma}\left(1 - \frac{\gamma}{t}\right)$ (from the mean value theorem), it suffices to show:

$$At^{-\gamma} - cA^p t^{-\gamma p} + Bt^{-\beta} \leq At^{-\gamma} - A\gamma t^{-\gamma-1}$$

which simplifies to:

$$-cA^p t^{-\gamma p} + Bt^{-\beta} \leq -A\gamma t^{-\gamma-1}$$

or equivalently:

$$cA^p t^{-\gamma p} \geq Bt^{-\beta} + A\gamma t^{-\gamma-1} \tag{1}$$

We now consider two cases based on the value of γ:

Case 1: $\gamma = \frac{1}{p-1}$

Then $\gamma p = \frac{p}{p-1}$ and since $\gamma = \min\left(\frac{1}{p-1}, \frac{\beta}{p}\right) = \frac{1}{p-1}$, we have $\frac{\beta}{p} \geq \frac{1}{p-1}$, i.e., $\beta \geq \frac{p}{p-1}$.

Thus:

- $t^{-\beta} \leq t^{-\gamma p}$ (since $\beta \geq \gamma p$)
- $t^{-\gamma-1} = t^{-\frac{p}{p-1}} = t^{-\gamma p}$

So inequality (1) becomes:

$$cA^p t^{-\gamma p} \geq (B + A\gamma)t^{-\gamma p}$$

which is satisfied if $cA^p \geq B + A\gamma$.

Case 2: $\gamma = \frac{\beta}{p}$

Then $\gamma p = \beta$ and since $\gamma = \min\left(\frac{1}{p-1}, \frac{\beta}{p}\right) = \frac{\beta}{p}$, we have $\frac{\beta}{p} \leq \frac{1}{p-1}$, i.e., $\beta \leq \frac{p}{p-1}$.

Thus:

- $t^{-\beta} = t^{-\gamma p}$

- $t^{-\gamma-1} = t^{-(\frac{\beta}{p}+1)} = o(t^{-\beta})$ since $\frac{\beta}{p} + 1 > \beta$ when $\beta \leq \frac{p}{p-1}$ and $p > 1$

For sufficiently large t, inequality (1) is satisfied if:

$$cA^p t^{-\gamma p} \geq Bt^{-\gamma p}$$

i.e., if $cA^p \geq B$.

In both cases, we can choose A sufficiently large to satisfy the required inequality. By induction, $y_t \leq g(t)$ for all $t \geq N_0$, which proves the result. □

Since $x_t \to 0$ (by Theorem 1), there exists N such that for all $t \geq N$, $x_t \leq x_0$. Therefore, for $t \geq N$, Assumption 2.8 gives us $f(x_t) \geq c_1 x_t^p$.

Case 1: $p = 1$

For $t \geq N$, the recurrence becomes:

$$x_{t+1} \leq x_t - c_1 x_t + \sigma_t^2 = (1 - c_1)x_t + \sigma_t^2$$

By Assumption 2.9, $\sigma_t^2 \leq Bt^{-\beta}$ for some $B > 0$. Applying Lemma A.3 with $\alpha = c_1$ and $b_t = \sigma_t^2$, we obtain:

$$x_t = O\left(\max\left((1 - c_1)^t, t^{-\beta}\right)\right) = O\left(\max\left(e^{-ct}, t^{-\beta}\right)\right)$$

where $c = -\log(1 - c_1) > 0$.

Case 2: $p > 1$

For $t \geq N$, the recurrence becomes:

$$x_{t+1} \leq x_t - c_1 x_t^p + \sigma_t^2$$

By Assumption 2.9, $\sigma_t^2 \leq Bt^{-\beta}$ for some $B > 0$. Applying Lemma A.4 with $c = c_1$ and $b_t = \sigma_t^2$, we obtain:

$$x_t = O\left(\max\left(t^{-\frac{1}{p-1}}, t^{-\frac{\beta}{p}}\right)\right)$$

This completes the proof of Theorem 2.10.

A.3. proof of Theorem 2.12

We have this lemma

Lemma A.5. *Suppose that $\widehat{\boldsymbol{\theta}} = \mathcal{M}(\mathcal{D})$ is an estimate of $\boldsymbol{\theta}$ with $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^t \sim \mathbb{P}_{\boldsymbol{\theta}}$ satisfying Assumption 2.11 with $r(t) = t^\kappa$. It then follows that*

$$\mathbb{E}[|\widehat{\theta}_i - \theta_i|^p] \leq \frac{pC_1}{\gamma} \cdot \Gamma\left(\frac{p}{\gamma}\right) \cdot C_2^{-p/\gamma} t^{-\frac{p\kappa}{\gamma}},$$

for any $i \in [p]$, where $\widehat{\theta}_i$ denotes the i-th element of $\widehat{\boldsymbol{\theta}}$.

Which means that when $\gamma, \kappa > 0$, and the estimate is unbiased, the Assumption 2.1 is satisfied. Then by Theorem 2.7, we can get for each $\delta > 0$,

$$\limsup_{t \to \infty} \mathbb{P}(\|\mathbf{e}_t\| > \delta) = 0.$$