Dataset Distillation with Neural Characteristic Function: A Minmax Perspective

Shaobo Wang[1,2] Yicun Yang[2] Zhiyuan Liu[2] Chenghao Sun[2]
Xuming Hu[3] Conghui He[4] Linfeng Zhang[1,2*]
[1]School of Artificial Intelligence, Shanghai Jiao Tong University
[2]EPIC Lab, Shanghai Jiao Tong University
[3]Hong Kong University of Science and Technology, Guangzhou
[4]Shanghai Artificial Intelligence Laboratory
{shaobowang1009,zhanglinfeng}@sjtu.edu.cn

Abstract

Dataset distillation has emerged as a powerful approach for reducing data requirements in deep learning. Among various methods, distribution matching-based approaches stand out for their balance of computational efficiency and strong performance. However, existing distance metrics used in distribution matching often fail to accurately capture distributional differences, leading to unreliable measures of discrepancy. In this paper, we reformulate dataset distillation as a minmax optimization problem and introduce Neural Characteristic Function Discrepancy (NCFD), a comprehensive and theoretically grounded metric for measuring distributional differences. NCFD leverages the Characteristic Function (CF) to encapsulate full distributional information, employing a neural network to optimize the sampling strategy for the CF's frequency arguments, thereby maximizing the discrepancy to enhance distance estimation. Simultaneously, we minimize the difference between real and synthetic data under this optimized NCFD measure. Our approach, termed Neural Characteristic Function Matching (NCFM), inherently aligns the phase and amplitude of neural features in the complex plane for both real and synthetic data, achieving a balance between realism and diversity in synthetic samples. Experiments demonstrate that our method achieves significant performance gains over state-of-the-art methods on both low- and high-resolution datasets. Notably, we achieve a 20.5% accuracy boost on ImageSquawk. Our method also reduces GPU memory usage by over 300× and achieves 20× faster processing speeds compared to state-of-the-art methods. To the best of our knowledge, this is the first work to achieve lossless compression of CIFAR-100 on a single NVIDIA 2080 Ti GPU using only 2.3 GB of memory.

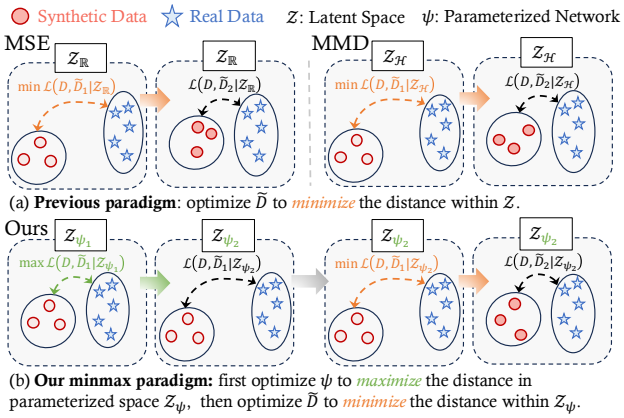

Figure 1. Comparison of different paradigms for dataset distillation. (a) The MSE approach compares point-wise features within Euclidean space, denoted as $\mathcal{Z}_\mathbb{R}$, while MMD evaluates moment differences in Hilbert space, $\mathcal{Z}_\mathcal{H}$. (b) Our method redefines distribution matching as a minmax optimization problem, where the distributional discrepancy is parameterized by a neural network ψ. We begin by optimizing ψ to maximize the discrepancy, thereby establishing the latent space $\mathcal{Z}_\psi$, and subsequently optimize the synthesized data $\tilde{\mathcal{D}}$ to minimize this discrepancy within $\mathcal{Z}_\psi$.

1. Introduction

Deep neural networks (DNNs) have achieved remarkable progress across a range of tasks, largely due to the availability of vast amounts of training data. However, training effectively with limited data remains challenging and crucial, particularly when large-scale datasets become too voluminous for storage. To address this, dataset distillation has been proposed to condense a large, real dataset into a smaller, synthetic one [6, 49, 52, 55, 56]. Dataset distillation has been applied in various areas, including neural architecture search [33, 44], continual learning [15, 51], medical image computing [29], and privacy protection [7, 8, 11].

Among dataset distillation methods, feature or distribution matching (DM) approaches [47, 55] have gained popu-

*Corresponding Author.

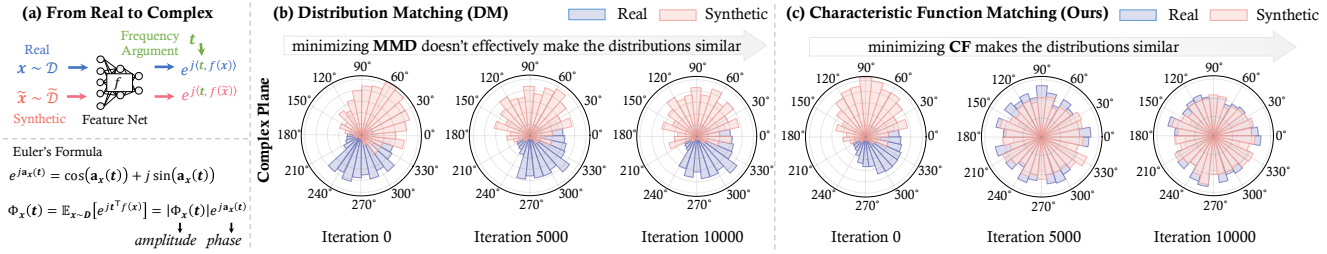

Figure 2. Comparison of different distribution matching methods. (a) Illustration of embedded features from the real domain to complex-plane features using Euler's formula [13]. The latent neural feature $\Phi_{\boldsymbol{x}}(\boldsymbol{t})$ captures the amplitude and phase information. (b) MMD-based methods align feature moments in the embedded domain but may not effectively align the overall distributions. (c) CF-based methods directly compare distributions by balancing the amplitude and phase in the complex plane, enhancing distributional similarity.

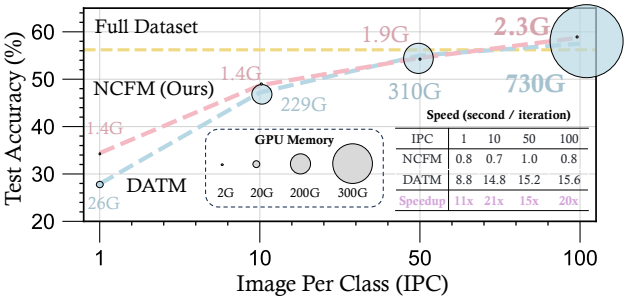

Figure 3. Comparison of performance, peak GPU memory usage, and distillation speed between the state-of-the-art (SOTA) distillation method and our NCFM on CIFAR-100 across various IPC values, evaluated on 8 NVIDIA H100 GPUs. Notably, NCFM reduces GPU memory usage by over 300×, achieves 20× faster distillation, and delivers better performance. We also successfully demonstrated lossless distillation using only 2.3GB GPU memory.

larity for their effective balance between high performance and computational efficiency. Unlike bi-level optimization-based distillation approaches [6, 20, 24, 54, 56], DM-based methods bypass the need for nested optimization. For instance, when learning with 50 images per class (IPC) on CIFAR-10 dataset, DM methods achieve higher test accuracy than gradient matching methods [24, 54, 56], while requiring only a tenth of the computation time.

A key challenge in DM lies in defining an effective metric to measure distributional discrepancies between real and synthetic datasets. Early methods primarily employed Mean Squared Error (MSE) to compare point-wise features [10, 38, 47], which operates in Euclidean space, $\mathcal{Z}_{\mathbb{R}}$, as illustrated on the left of Figure 1(a). However, MSE directly matches pixel-level or patch-level information without capturing the semantic structures embedded in high-dimensional manifolds, which falls short for distribution comparison. Later methods [53, 55, 57] employ Maximum Mean Discrepancy (MMD) as a metric. Nevertheless, research in generative modeling [4, 25] has shown that MMD aligns moments of distributions in a latent Hilbert space, $\mathcal{Z}_{\mathcal{H}}$, as shown on the right of Figure 1(a). While distributional equivalence implies moment equivalence, the converse is not necessarily true: aligning moments alone does not guarantee full distributional matching. As illustrated in Figure 2(b), MMD-based methods may fail to capture overall distributional alignment between real and synthetic data, resulting in suboptimal synthesized image quality.

To overcome these limitations, we propose a novel approach that reformulates distribution matching as an *adversarial minmax optimization problem*, as depicted in Figure 1(b). By leveraging the minmax paradigm, we adaptively learn the discrepancy metric, enabling it to *maximize* the separability between real and synthetic data distributions. This dynamic adjustment addresses the rigidity of fixed metrics like MSE and MMD. Meanwhile, the synthetic data is iteratively optimized to *minimize* the dynamically refined discrepancy measure. Building upon this foundation, we introduce Neural Characteristic Discrepancy (NCFD), a parameterized metric based on the Characteristic Function (CF), which provides a precise and comprehensive representation of the underlying probability distribution. Defined as the Fourier transform of the probability density function, the CF encapsulates all relevant information about a distribution [3, 5, 14, 21, 31, 41]. The CF offers a one-to-one correspondence with the cumulative density function, ensuring the robustness and reliability.

In our framework, an auxiliary network embeds features while a lightweight sampling network is optimized to dynamically adjust its CF sampling strategy using a scale mixture of normals. During the distillation process, we iteratively minimize the NCFD to bring synthetic data closer to real data, while training the sampling network to maximize NCFD, thereby improving the metric's robustness and accuracy. Unlike MMD which has quadratic computational complexity, NCFD achieves linear time computational complexity. Our method, Neural Characteristic Function Matching (NCFM), aligns both the phase and amplitude of neural features in the complex plane, achieving a balanced synthesis of realism and diversity in the generated images. As shown in Figure 2(c), NCFM effectively captures overall distributional information, leading to well-aligned synthetic and real data distributions after optimization. Our contributions are as follows:

1. We reformulate the distribution matching problem as a minmax optimization problem, where the sampling net-

work maximizes the distributional discrepancy to learn a proper discrepancy metric, while the synthesized images are optimized to minimize such discrepancy.

2. We introduce Neural Characteristic Function Matching (NCFM), which aligns the phase and amplitude information of neural features in the complex plane for both real and synthetic data, achieving a balance between realism and diversity in synthetic data.
3. Extensive experiments across multiple benchmark datasets demonstrate the superior performance and efficiency of NCFM. Particularly, on high-resolution datasets, NCFM achieves significant accuracy gains of up to 20.5% on ImageSquawk and 17.8% on ImageMeow at 10 IPC compared to SOTA methods.
4. NCFM achieves unprecedented efficiency in computational resources. As shown in Figure 3, our method dramatically reduces resource requirements with better performance, **achieving more than 300× reduction in GPU memory usage compared with DATM [16]**. Most remarkably, **NCFM demonstrates lossless dataset distillation on both CIFAR-10 and CIFAR-100 using about merely 2GB GPU memory**, enabling all experiments to be conducted on a single NVIDIA 2080 Ti GPU.

2. Related Work

Dataset Distillation Methods Based on Distribution and Feature Matching. Dataset distillation was proposed by [49]. Compared with various bi-level DD methods, DM [55] is regarded as a efficient method that balances the performance and computational efficiency, without involving the nested model optimization. These methods can be classified into two directions, *i.e.*, point-wise and moment-wise matching. For moment-wise matching, DM-based methods [53, 55, 57] propose to minimize the maximum mean discrepancy (MMD) between synthetic and real datasets. For point-wise feature matching, they typically design better strategies to match features extracted across layers in convolutional neural networks, and apply further adjustments to improve the performance [10, 38, 47]. However, moment-based and point-based matching methods may not capture the overall distributional discrepancy between synthetic and real data, as they are not sufficient conditions for distributional equivalence.

Characteristic Function as a Distributional Metric. The characteristic function is a unique and universal metric for measuring distributional discrepancy, defined as the Fourier transform of the probability density function [3]. The CF of any real-valued random variable completely defines its probability distribution, providing an alternative analytical approach compared to working directly with probability density or cumulative distribution functions. Unlike the moment-generating function, the CF always exists when treated as a function of a real-valued argument [5]. Recently, the CFD has been adopted in deep learning for various tasks, *e.g.*, several works have been proposed to use the CFD for generative modeling [1, 27]. However, none of prior works have considered the CFD for distillation.

3. Preliminaries: Distribution Matching

Distribution Matching (DM) was first introduced by [55] as an alternative to traditional bi-level optimization techniques, such as gradient matching methods [20, 24, 54, 56] and trajectory matching methods [6, 9, 12, 16]. Classical DM approaches focus on minimizing the discrepancy between the distributions of real and synthetic data, typically categorized into two main types: feature point matching and moment matching. Feature point matching methods [10, 38, 47] directly compare point-wise features using Mean Square Error (MSE), as defined by:

$$\mathcal{L}_{\text{MSE}} = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\tilde{\boldsymbol{x}}\sim\tilde{\mathcal{D}}}\left[\left\|f(\boldsymbol{x}) - f(\tilde{\boldsymbol{x}})\right\|^2\right], \tag{1}$$

where f denotes the feature extractor network, $\mathcal{D}$ and $\tilde{\mathcal{D}}$ represent the real and synthetic data distributions, respectively, $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ are samples drawn from $\mathcal{D}$ and $\tilde{\mathcal{D}}$. However, MSE may not be ideal for comparing distributions, as it only considers direct feature comparisons in Euclidean space, neglecting important semantic information.

In another line, notable works employed Maximum Mean Discrepancy (MMD) to align high-order moments in the latent feature space [53, 55, 57]. Rigorously, MMD is defined to match moments within the Reproducing Kernel Hilbert Space (RKHS) induced by a selected kernel function. The MMD loss can be formulated as:

$$\begin{aligned}&\sup_{f\in\mathcal{F}} \left\|\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[f(\boldsymbol{x})\right] - \mathbb{E}_{\tilde{\boldsymbol{x}}\sim\tilde{\mathcal{D}}}\left[f(\tilde{\boldsymbol{x}})\right]\right\|^2, \\ &= \sup_{f\in\mathcal{F}} \left(\mathcal{K}_{\mathcal{D},\mathcal{D}} + \mathcal{K}_{\tilde{\mathcal{D}},\tilde{\mathcal{D}}} - 2\mathcal{K}_{\mathcal{D},\tilde{\mathcal{D}}}\right),\end{aligned} \tag{2}$$

where $\mathcal{K}_{\mathcal{D},\tilde{\mathcal{D}}} = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\tilde{\boldsymbol{x}}\sim\tilde{\mathcal{D}}}[\mathcal{K}_{f(\boldsymbol{x}),f(\tilde{\boldsymbol{x}})}]$ denotes the kernel function associated with feature extractor f in function class $\mathcal{F}$. The choice of kernel function $\mathcal{K}$ is not unique and requires careful selection for MMD to be effective. However, instead of selecting certain kernel function, most DM-based methods [10, 55, 57] align moments directly in the feature space, commonly approximated as:

$$\mathcal{L}_{\text{MMD}} = \left\|\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left[f(\boldsymbol{x})\right] - \mathbb{E}_{\tilde{\boldsymbol{x}}\sim\tilde{\mathcal{D}}}\left[f(\tilde{\boldsymbol{x}})\right]\right\|^2. \tag{3}$$

We argue that such empirical MMD estimates lack rigor, as they do not provide a maximal upper bound on the discrepancy, falling short of MMD's theoretical requirements.

4. Adversarial Distribution Matching

4.1. Minmax Framework

To address existing challenges with discrepancy measure selection, we propose a new approach that reformulates distribution matching as a minmax optimization problem. In

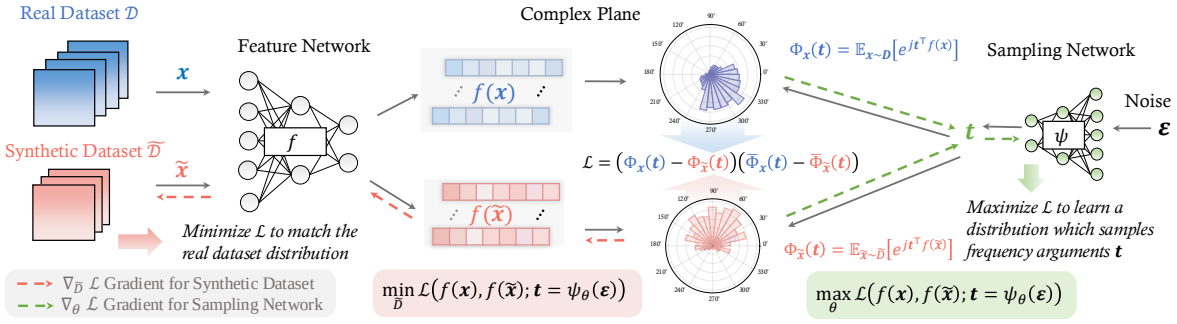

Figure 4. Dataset Distillation with Neural Characteristic Function Matching (NCFM). Real and synthetic data points are sampled and embedded through a feature extractor network. The synthetic data is optimized by minimizing the distributional discrepancy between real and synthetic data, measured via the Neural Characteristic Function Discrepancy (NCFD) in the complex plane. Additionally, an auxiliary network learns an optimal sampling distribution for the frequency arguments of the characteristic function. Best viewed in color.

this framework, we maximize the discrepancy measure to define a robust discrepancy metric, parameterized by a neural network ψ. Concurrently, we minimize the discrepancy between the synthetic dataset $\tilde{\mathcal{D}}$ and the real dataset $\mathcal{D}$ by optimizing the synthetic data distribution $\tilde{\mathcal{D}}$. Formally, this minmax optimization problem is expressed as:

$$\min_{\tilde{\mathcal{D}}} \max_{\psi} \mathcal{L}(\tilde{\mathcal{D}}, \mathcal{D}, f, \psi), \tag{4}$$

where $\mathcal{L}$ denotes the discrepancy measure, f is the feature extractor network, and ψ is the network learning the discrepancy metric. This minmax framework seeks the optimal synthetic data distribution $\tilde{\mathcal{D}}$ that minimizes $\mathcal{L}$ while network ψ maximizes $\mathcal{L}$ to establish a robust metric.

4.2. Neural Characteristic Function Matching

4.2.1. Neural Characteristic Function Discrepancy

To define a suitable discrepancy metric within this minmax framework, we propose a novel discrepancy measure based on the characteristic function (CF), which enables direct and robust assessment of distributional discrepancies. Characteristic functions are a mainstay in probability theory, often used as an alternative to probability density functions due to their unique properties. Specifically, the CF of a random variable $\boldsymbol{x}$ is the expectation of its complex exponential transform, defined as:

$$\Phi_{\boldsymbol{x}}(\boldsymbol{t}) = \mathbb{E}_{\boldsymbol{x}}\left[e^{j\langle \boldsymbol{t}, \boldsymbol{x}\rangle}\right] = \int_{\boldsymbol{x}} e^{j\langle \boldsymbol{t}, \boldsymbol{x}\rangle} dF(\boldsymbol{x}), \tag{5}$$

where $F(\boldsymbol{x})$ denotes the cumulative distribution function (cdf) of $\boldsymbol{x}$, $j = \sqrt{-1}$, and $\boldsymbol{t}$ is the frequency argument. Since the cdf is not directly accessible in practice, we approximate the CF empirically as $\Phi_{\boldsymbol{x}}(\boldsymbol{t}) = \frac{1}{N}\sum_{i=1}^{N} e^{j\langle \boldsymbol{t}, \boldsymbol{x}_i\rangle}$, where N is the sample size in the dataset. The CF provides a theoretically principled description of a distribution, summarized in the following theorems.

Theorem 1 (Lévy's Convergence Theorem [31]) *Let $\{X_n\}_{n=1}^{\infty}$ be a sequence of random variables with characteristic functions $\Phi_n(\boldsymbol{t}) = \mathbb{E}_{X_n}\left[e^{j\langle \boldsymbol{t}, X_n\rangle}\right]$. Suppose $\Phi_n(\boldsymbol{t}) \to \Phi(\boldsymbol{t})$ pointwise for each $\boldsymbol{t} \in \mathbb{R}^d$ as $n \to \infty$. If $\Phi(\boldsymbol{t})$ is continuous at $\boldsymbol{t} = 0$, then there exists a random variable X with characteristic function $\Phi(\boldsymbol{t})$, and X_n converges in distribution to X.*

Theorem 2 (Uniqueness for Characteristic Functions [14]) *If two random variables X and Y have the same characteristic function, $\Phi_X(\boldsymbol{t}) = \Phi_Y(\boldsymbol{t})$ for all $\boldsymbol{t}$, then X and Y are identically distributed. In other words, a characteristic function uniquely determines the distribution.*

By Theorems 1 and 2, the empirical CF weakly converges to the population CF, ensuring that the CF serves as a reliable proxy for the underlying distribution. Based on the CF, we define our characteristic function discrepancy (CFD) as:

$$\mathcal{C}_{\mathcal{T}}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \int_{\boldsymbol{t}} \sqrt{\underbrace{(\Phi_{\boldsymbol{x}}(\boldsymbol{t}) - \Phi_{\tilde{\boldsymbol{x}}}(\boldsymbol{t}))(\bar{\Phi}_{\boldsymbol{x}}(\boldsymbol{t}) - \bar{\Phi}_{\tilde{\boldsymbol{x}}}(\boldsymbol{t}))}_{\mathrm{Chf}(\boldsymbol{t})}}\, dF_{\mathcal{T}}(\boldsymbol{t}), \tag{6}$$

where $\bar{\Phi}$ is the complex conjugate of Φ, and $F_{\mathcal{T}}(\boldsymbol{t})$ is the CDF of a sampling distribution on $\boldsymbol{t}$. To simplify, we let $\mathrm{Chf}(\boldsymbol{t}) = (\Phi_{\boldsymbol{x}}(\boldsymbol{t}) - \Phi_{\tilde{\boldsymbol{x}}}(\boldsymbol{t}))(\bar{\Phi}_{\boldsymbol{x}}(\boldsymbol{t}) - \bar{\Phi}_{\tilde{\boldsymbol{x}}}(\boldsymbol{t}))$ for further analysis. We show that $\mathcal{C}_{\mathcal{T}}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ defines a valid distance metric in the following theorem.

Theorem 3 (CFD as a Distance Metric.) *The CF discrepancy $\mathcal{C}_{\mathcal{T}}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$, as defined in Eq. (6), serves as a distance metric between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ when the support of $\mathcal{T}$ resides in Euclidean space. It satisfies the properties of non-negativity, symmetry, and the triangle inequality.*

Theorem 3 establishes that CFD is a valid distance metric. Furthermore, we demonstrate that this formulation decomposes CFD into phase, $\boldsymbol{a}_{\boldsymbol{x}}(\boldsymbol{t})$, and amplitude, $|\Phi_{\boldsymbol{x}}(\boldsymbol{t})|$, com-

ponents through Euler's formula:

$$\begin{aligned}\text{Chf}(\boldsymbol{t}) &= |\Phi_{\boldsymbol{x}}(\boldsymbol{t})|^2 + |\Phi_{\tilde{\boldsymbol{x}}}(\boldsymbol{t})|^2 \\ &- |\Phi_{\boldsymbol{x}}(\boldsymbol{t})|\,|\Phi_{\tilde{\boldsymbol{x}}}(\boldsymbol{t})|\,(2\cos(\boldsymbol{a}_{\boldsymbol{x}}(\boldsymbol{t}) - \boldsymbol{a}_{\tilde{\boldsymbol{x}}}(\boldsymbol{t}))) \\ &= \underbrace{\left(|\Phi_{\boldsymbol{x}}(\boldsymbol{t}) - \Phi_{\tilde{\boldsymbol{x}}}(\boldsymbol{t})|\right)^2}_{\text{amplitude difference}} \\ &+ 2\,|\Phi_{\boldsymbol{x}}(\boldsymbol{t})|\,|\Phi_{\tilde{\boldsymbol{x}}}(\boldsymbol{t})|\,\underbrace{(1 - \cos(\boldsymbol{a}_{\boldsymbol{x}}(\boldsymbol{t}) - \boldsymbol{a}_{\tilde{\boldsymbol{x}}}(\boldsymbol{t})))}_{\text{phase difference}},\end{aligned} \tag{7}$$

• *Phase Information:* the term $1 - \cos(\boldsymbol{a}_{\boldsymbol{x}}(\boldsymbol{t}) - \boldsymbol{a}_{\tilde{\boldsymbol{x}}}(\boldsymbol{t}))$ represents the phase, encoding data centres crucial for *realism*.
• *Amplitude Information:* the term $|\Phi_{\boldsymbol{x}}(\boldsymbol{t}) - \Phi_{\tilde{\boldsymbol{x}}}(\boldsymbol{t})|^2$ captures the distribution scale, contributing to the *diversity*.

This phase-amplitude decomposition, supported in signal processing [32, 35] and generative models [27], provides insight into CFD's descriptive power. In practice, to reduce computational cost, we furehr introduce an additional feature extractor f to map input variables into a latent space, which is similar to previous works in distribution matching [10, 26, 55, 57]. We also use a parameterized sampling network ψ to obtain the distribution of frequency argument $\boldsymbol{t}$, thereby extending the CFD to a more general parameterized form, *i.e.*, *Neural Characteristic Function Discrepancy (NCFD)* as $\mathcal{C}_{\mathcal{T}}(\boldsymbol{x}, \tilde{\boldsymbol{x}}; f, \psi) = \int_{\boldsymbol{t}} \sqrt{\text{Chf}(\boldsymbol{t}; f)} dF_{\mathcal{T}}(\boldsymbol{t}; \psi)$, where $\text{Chf}(\boldsymbol{t}; f)$ is defined as $\left(\left|\Phi_{f(\boldsymbol{x})}(\boldsymbol{t}) - \Phi_{f(\tilde{\boldsymbol{x}})}(\boldsymbol{t})\right|\right)^2 + 2\left|\Phi_{f(\boldsymbol{x})}(\boldsymbol{t})\right|\left|\Phi_{f(\tilde{\boldsymbol{x}})}(\boldsymbol{t})\right|(1 - \cos(\boldsymbol{a}_{f(\boldsymbol{x})}(\boldsymbol{t}) - \boldsymbol{a}_{f(\tilde{\boldsymbol{x}})}(\boldsymbol{t})))$.

4.2.2. Determining the sampling strategy for NCFD

The core aspect in optimizing $\mathcal{C}_{\mathcal{T}}(\boldsymbol{x}, \tilde{\boldsymbol{x}}; f, \psi)$ lies in determining the form of $F_{\mathcal{T}}(\boldsymbol{t}; \psi)$, *i.e.*, how to correctly and efficiently sample $\boldsymbol{t}$ from a carefully picked distribution. Similar with works in generative adversarial network [1, 28], we define $F_{\mathcal{T}}(\boldsymbol{t})$ as the cumulative distribution function (cdf) of a *scale mixture of normals*, as $p_{\mathcal{T}}(\boldsymbol{t}) = \int_{\boldsymbol{\Sigma}} \boldsymbol{\mathcal{N}}(\boldsymbol{t}|\boldsymbol{0}, \boldsymbol{\Sigma}) p_{\boldsymbol{\Sigma}}(\boldsymbol{\Sigma}) d\boldsymbol{\Sigma}$, where $p_{\mathcal{T}}(\boldsymbol{t})$ is the probability density function (pdf) of $F_{\mathcal{T}}(\boldsymbol{t})$, $\boldsymbol{\mathcal{N}}(\boldsymbol{t}|\boldsymbol{0}, \boldsymbol{\Sigma})$ denotes a zero-mean Gaussian distribution with covariance $\boldsymbol{\Sigma}$, and $p_{\boldsymbol{\Sigma}}(\boldsymbol{\Sigma})$ represents the distribution of $\boldsymbol{\Sigma}$. We observe that as the number of sampled frequency arguments increases, the approximation of the empirical CF improves, as guaranteed by Lévy's Convergence Theorem [31], ultimately leading to higher quality synthetic data.

4.2.3. Distribution Matching with NCFD

Given the NCFD measure $\mathcal{C}_{\mathcal{T}}(\boldsymbol{x}, \tilde{\boldsymbol{x}}; f, \psi)$, we now propose a method to utilize NCFD for distribution matching, termed as Neural Characteristic Function Matching (NCFM). A visual illustration of the NCFM pipeline is provided in Figure 4. On one hand, we maximize the NCFD to learn an effective discrepancy metric by optimizing the network ψ. On the other hand, we minimize this learned NCFD to obtain an optimal synthetic dataset, $\tilde{\mathcal{D}}$. In practice, we introduce a hyper-parameter α to balance the amplitude and phase information in the NCFD, then the minmax optimization problem can be formulated as:

$$\begin{aligned}&\min_{\tilde{\mathcal{D}}} \max_{\psi} \mathcal{L}(\tilde{\mathcal{D}}, \mathcal{D}, f, \psi) = \min_{\tilde{\mathcal{D}}} \max_{\psi} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}, \tilde{\boldsymbol{x}}\sim\tilde{\mathcal{D}}} \mathcal{C}_{\mathcal{T}}(\boldsymbol{x}, \tilde{\boldsymbol{x}}; f, \psi) \\ &= \min_{\tilde{\mathcal{D}}} \max_{\psi} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}, \tilde{\boldsymbol{x}}\sim\tilde{\mathcal{D}}} \int_{\boldsymbol{t}} \sqrt{\text{Chf}(\boldsymbol{t}; f)}\, dF_{\mathcal{T}}(\boldsymbol{t}; \psi) \\ &\text{where} \quad \text{Chf}(\boldsymbol{t}; f) = \alpha \left(\left(\left|\Phi_{f(\boldsymbol{x})}(\boldsymbol{t}) - \Phi_{f(\tilde{\boldsymbol{x}})}(\boldsymbol{t})\right|\right)^2\right) + (1 - \alpha) \cdot \\ &(2\left|\Phi_{f(\boldsymbol{x})}(\boldsymbol{t})\right|\left|\Phi_{f(\tilde{\boldsymbol{x}})}(\boldsymbol{t})\right|) \cdot (1 - \cos(\boldsymbol{a}_{f(\boldsymbol{x})}(\boldsymbol{t}) - \boldsymbol{a}_{f(\tilde{\boldsymbol{x}})}(\boldsymbol{t}))).\end{aligned} \tag{8}$$

For the design of f, we used a hybrid approach that combines a pre-trained model with a randomly initialized model, both selected from a subset of trained models. This ensures that the feature extractor remains moderately diverse yet discriminative. The hybrid feature extractor is constructed by β-blending the checkpoints of the initial and final models, where each model is chosen from a specific subset of available models. At each distillation step, the blending coefficient $\beta \in (0, 1)$ is sampled from a uniform distribution $\mathcal{U}(0, 1)$, providing a balanced combination of initial and final checkpoints. Our NCFM can be seamlessly integrated with additional data curation steps, such as generating soft labels with a pre-trained neural network and performing dataset finetuning. Unlike prior methods that focus on learning soft labels [16, 19, 36], NCFM simply leverages a pre-trained network to efficiently generate soft labels for the distilled dataset, improving both efficiency and effectiveness. However, these additional curation steps are not essential for NCFM, as it can achieve SOTA performance within the pure minmax framework.

5. Experiments

5.1. Setup

Baseline methods. We compared NCFM with several representative approaches in dataset distillation and coreset selection. These include gradient-matching methods such as DC [56], DCC [24], DSA and DSAC [54]. Kernel-based methods like KIP [34] and FrePo [58] were also included. Distribution-matching methods like CAFE [47], DM [55], IDM [57], M3D [53], IID [10], and DSDM [26] were part of the evaluation. We also included trajectory-matching methods such as MTT [6], FTD [12], ATT [30], and TESLA [9]. *State-of-the-art* methods like DATM [16], G-VBSM [40], and RDED [45] were also considered in our comparisons. Additionally, we benchmarked our method against classical coreset selection techniques, including random selection, Herding [50], and Forgetting [46].

Datasets and Networks. Our evaluations were conducted on widely-used datasets, including CIFAR-10 and CIFAR-100 [22] with resolution of 32×32, Tiny ImageNet [23] with resolution of 64×64, and ImageNet subsets with resolution of 128×128, *i.e.*, ImageNette, ImageWoof, ImageFruit, Im-

Table 1. Results of NCFM on CIFAR-10/100, and Tiny ImageNet (resolution of 64×64) datasets.

Dataset	CIFAR-10			CIFAR-100			Tiny ImageNet		
IPC	1	10	50	1	10	50	1	10	50
Ratio (%)	0.02	0.2	1	0.2	2	10	0.2	2	10
Random	14.4±2.0	26.0±1.2	43.4±1.0	4.2±0.3	14.6±0.5	30.0±0.4	1.4±0.1	5.0±0.2	15.0±0.4
Herding	21.5±1.2	31.6±0.7	40.4±0.6	8.4±0.3	17.3±0.3	33.7±0.5	2.8±0.2	6.3±0.2	16.7±0.3
Forgetting	13.5±1.2	23.3±1.0	23.3±1.1	4.5±0.2	15.1±0.3	30.5±0.3	1.6±0.1	5.1±0.2	15.0±0.3
DC	28.3±0.5	44.9±0.5	53.9±0.5	12.8±0.3	25.2±0.3	-	-	-	-
DSA	28.8±0.7	52.1±0.5	60.6±0.5	13.9±0.3	32.3±0.3	42.8±0.4	-	-	-
DCC	32.9±0.8	49.4±0.5	61.6±0.4	13.3±0.3	30.6±0.4	40.0±0.3	-	-	-
DSAC	34.0±0.7	54.5±0.5	64.2±0.4	14.6±0.3	14.6±0.3	39.3±0.4	-	-	-
FrePo	46.8±0.7	65.5±0.4	71.7±0.2	28.7±0.1	42.5±0.2	44.3±0.2	15.4±0.3	25.4±0.2	-
MTT	46.3±0.8	65.3±0.7	71.6±0.2	24.3±0.3	40.1±0.4	47.7±0.2	8.8±0.3	23.2±0.2	28.0±0.3
ATT	48.3±1.0	67.7±0.6	74.5±0.4	26.1±0.3	44.2±0.5	51.2±0.3	11.0±0.5	25.8±0.4	-
FTD	46.8±0.3	66.6±0.3	73.8±0.2	25.2±0.2	43.4±0.3	48.5±0.3	10.4±0.3	24.5±0.2	-
TESLA	48.5±0.8	66.4±0.8	72.6±0.7	24.8±0.4	41.7±0.3	47.9±0.3	-	-	-
CAFE	30.3±1.1	46.3±0.6	55.5±0.6	12.9±0.3	27.8±0.3	37.9±0.3	-	-	-
DM	26.0±0.8	48.9±0.6	63.0±0.4	11.4±0.3	29.7±0.3	43.6±0.4	3.9±0.2	12.9±0.4	24.1±0.3
IDM	45.6±0.7	58.6±0.1	67.5±0.1	20.1±0.3	45.1±0.1	50.0±0.2	10.1±0.2	21.9±0.2	27.7±0.3
M3D	45.3±0.3	63.5±0.2	69.9±0.5	26.2±0.3	42.4±0.2	50.9±0.7	-	-	-
IID	47.1±0.1	59.9±0.1	69.0±0.3	24.6±0.1	45.7±0.4	51.3±0.4	-	-	-
DSDM	45.0±0.4	66.5±0.3	75.8±0.3	19.5±0.2	46.2±0.3	54.0±0.2	-	-	-
G-VBSM	-	46.5±0.7	54.3±0.3	16.4±0.7	38.7±0.2	45.7±0.4	-	-	-
NCFM (Ours)	**49.5**±0.3	**71.8**±0.3	**77.4**±0.3	**34.4**±0.5	**48.7**±0.3	**54.7**±0.2	**18.2**±0.5	**26.8**±0.6	**29.6**±0.5
Whole Dataset		84.8±0.1			56.2±0.3			37.6±0.4	

ageMeow, ImageSquawk, and ImageYellow [18]. Following prior studies [16, 48], we used networks with instance normalization as the default setting. Specifically, dataset distillation is performed with a 3-layer ConvNet for CIFAR-10/100, a 4-layer ConvNet for Tiny ImageNet, and a 5-layer ConvNet for ImageNet subsets. All experiments were conducted with 10 evaluations for fairness, primarily using a single NVIDIA 4090 GPU.

Other Settings. Following prior works, we implemented differential augmentation [47, 54] and applied multi-formation parameterization with a scale factor of $\rho = 2$ for images, as in [20, 57]. We employed AdamW as our optimizer. In our setup, we set the number of sampled frequency arguments to 1024. The number of mixture Gaussian components in the sampling network is set to the number of frequency arguments divided by 16, balancing the sampling network diversity and computational efficiency. Further details are provided in the supplementary material.

5.2. Main Results

We verified the effectiveness of NCFM on various benchmark datasets of different image-per-class (IPC) settings[1].

CIFAR-10/100 and Tiny ImageNet. As shown in Table 1, NCFM outperforms all state-of-the-art (SOTA) baselines. Specifically, it surpasses distribution matching methods using traditional metrics like MSE and MMD, achieving improvements of 23.5% and 23.0% on CIFAR-10 and CIFAR-100 with 1 IPC compared to DM [55]. Additionally, NCFM maintains SOTA performance even against computationally intensive methods like MTT [6]. Results for larger IPC settings and comparisons with other SOTA methods like DATM [16] are in the supplementary material.

Higher-resolution Datasets. We also evaluated NCFM on larger datasets, specifically the ImageNet subsets. As shown in Table 2, NCFM demonstrates strong performance across these challenging benchmarks. In 10 IPC setting, our method achieves substantial improvements of 20.5% on ImageSquawk, compared to the baseline MTT [6]. Remarkably, NCFM exhibits robust performance under relatively small IPC. For instance, compared to RDED [45], NCFM yields a significant improvement of 19.6% on ImageNette.

Computational Efficiency Evaluation. We tested the training speed and GPU memory of our NCFM compared with *strong* baseline methods on different datasets. As conventional recognition, trajectory matching based methods usually achieve better results than distribution matching in practice [6, 9, 12]. However, both superior training efficiency and GPU memory efficiency are observed in NCFM across all benchmark datasets, while achieving better results. Specifically, we measured the average training time over 1000 distillation iterations for each method, as summarized in Table 3. For CIFAR-100 at IPC 50, NCFM achieves nearly 30× faster speeds compared to TESLA [9] without the sampling network, and maintains over 20× faster speeds with the sampling network included. Moreover, we conducted a comprehensive analysis of computational efficiency, where GPU memory is expressed as the peak memory usage during 1000 iterations of training, as shown in Table 3. While most existing methods encounter out of memory (OOM) issues at IPC = 50, our method requires only

[1]We provide further results on continual learning, neural architecture search, and larger IPC datasets in the supplementary material.

Table 2. Results on ImageNet subsets (resolution of 128×128) when employing NCFM across different IPCs.

Dataset	ImageNette		ImageWoof		ImageFruit		ImageMeow		ImageSquawk		ImageYellow	
IPC	1	10	1	10	1	10	1	10	1	10	1	10
Ratio (%)	0.105	1.050	0.110	1.100	0.077	0.77	0.077	0.77	0.077	0.77	0.077	0.77
Random	23.5±4.8	47.7±2.4	14.2±0.9	27.0±1.9	13.2±0.8	21.4±1.2	13.8±0.6	29.0±1.1	21.8±0.5	40.2±0.4	20.4±0.6	37.4±0.5
MTT	47.7±0.9	63.0±1.3	**28.6±0.8**	35.8±1.8	26.6±0.8	40.3±1.3	30.7±1.6	40.4±2.2	39.4±1.5	52.3±1.0	45.2±0.8	60.0±1.5
DM	32.8±0.5	58.1±0.3	21.1±1.2	31.4±0.5	-	-	-	-	31.2±0.7	50.4±1.2	-	-
RDED	33.8±0.8	63.2±0.7	18.5±0.9	40.6±2.0	-	-	-	-	-	-	-	-
NCFM (Ours)	**53.4±1.6**	**77.6±1.0**	27.2±1.1	**48.4±1.3**	**29.2±0.7**	**44.8±1.5**	**34.6±0.7**	**58.2±1.2**	**41.6±1.2**	**72.8±0.9**	**46.6±1.5**	**74.2±1.4**
Whole Dataset	87.4±1.0		67.0±1.3		63.9±2.0		66.7±1.1		87.5±0.3		84.4±0.6	

Table 3. Training speed (s/iter) and peak GPU memory (GB) comparison on a single NVIDIA A100 80G. OOM marks out-of-memory cases. 'Reduction' shows NCFM's speed and memory improvements over the best-performing baseline in the table.

Resource	Speed (s/iter)				GPU Memory (GB)			
Dataset	CIFAR-100		Tiny ImageNet		CIFAR-100		Tiny ImageNet	
IPC	10	50	10	50	10	50	10	50
MTT	1.92	OOM	OOM	OOM	61.6	OOM	OOM	OOM
FTD	1.68	OOM	OOM	OOM	61.4	OOM	OOM	OOM
TESLA	5.71	28.24	42.01	OOM	10.3	44.2	69.6	OOM
DATM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
NCFM w/o ψ	0.73	0.96	2.40	5.67	1.4	1.9	6.4	8.4
Reduction	**2.3**×	**29.4**×	**17.5**×	-	**7.4**×	**23.3**×	**10.9**×	-
NCFM	1.33	1.36	3.27	7.22	1.6	2.0	6.5	8.7
Reduction	**1.3**×	**20.8**×	**12.8**×	-	**6.4**×	**22.1**×	**10.7**×	-

about 1.9GB GPU memory on CIFAR-100. This further demonstrates the exceptional scalability of our approach under high IPC conditions. Further results on CIFAR-10 are provided in the supplementary material.

Cross-Architecture Generalization. We evaluated the cross-architecture generalization capability of our method by testing its performance on various network architectures, including AlexNet [22], VGG-11 [42], and ResNet-18 [17]. In this evaluation, synthetic data were condensed using a 3-layer ConvNet, and each method was subsequently tested across different architectures to assess robustness and adaptability. Tables 4 summarize the results on CIFAR-10 with 10 and 50 IPC settings, respectively. In both cases, NCFM consistently outperformed other methods across all architectures, demonstrating its strong ability to generalize effectively even when trained on a different architecture. Results on other backbone networks beyond ConvNet are provided in the supplementary material.

5.3. Ablation Study

5.3.1. Effect of the Sampling Network

To rigorously evaluate the impact of the sampling network, ψ, within the minmax paradigm of NCFM, we conducted performance comparisons with and without this component. To ensure a controlled and fair assessment, no additional data curation techniques were applied (such as fine-tuning or soft label integration). As shown in Table 5, employing the sampling network ψ yields substantial improvements in

Table 4. Cross-architecture generalization performance (%) on CIFAR-10. The synthetic data is condensed using ConvNet, and each method is evaluated on different architectures.

IPC	Method	ConvNet	AlexNet	VGG	ResNet
10	DSA	52.1±0.4	35.9±1.3	43.2±0.5	35.9±1.3
	MTT	64.3±0.7	34.2±2.6	50.3±0.8	34.2±2.6
	KIP	47.6±0.9	24.4±3.9	42.1±0.4	24.4±3.9
	NCFM	**71.8±0.3**	**67.9±0.5**	**68.0±0.3**	**67.7±0.5**
50	DSA	59.9±0.8	53.3±0.7	51.0±1.1	47.3±1.0
	DM	65.2±0.4	61.3±0.6	59.9±0.8	57.0±0.9
	NCFM	**77.4±0.3**	**75.5±0.3**	**75.5±0.3**	**73.8±0.2**

synthetic data quality across various datasets. For example, integrating ψ into our method provides a 3.2% performance increase on CIFAR-10 at 50 IPC. Our method yields a 2.6% performance increase on Tiny ImageNet at 1 IPC and 10.1% at 10 IPC. Similar trends are observed across ImageNet subsets, including gains of 2.8% on ImageMeow and 2.0% on ImageSquawk. The strong performance benefits from sampling network ψ emphasize the effectiveness of the minmax paradigm compared to straightforward CFD minimization.

5.3.2. Impact of Amplitude and Phase Components

We examine individual contributions of amplitude and phase alignment within the NCFD measure. By selectively adjusting amplitude or phase alignment, controlled by the hyperparameter α that represents the ratio of amplitude to phase weight in the loss function, we find that both components are essential. To further evaluate the effect of α on performance, we conducted ablation studies on the CIFAR-10 and CIFAR-100 datasets. As noted in prior works [32, 35], the amplitude term primarily enhances the diversity of generated data, while the phase term contributes to realism by accurately capturing data centers. For example, as shown in Figure 5, on CIFAR-10 with 10 IPC, when the amplitude information dominates the loss (*e.g.*, $\alpha = 0.999$), the test accuracy decreases about 3% compared to our best results. Conversely, when the phase information dominates (*e.g.*, $\alpha = 0.001$), the test accuracy decreases by about 1%. Results demonstrate that a balanced integration of both components yields the highest accuracy.

Table 5. Test Performance (%) on CIFAR-10, CIFAR-100, Tiny ImageNet and ImageNet subsets with and without the sampling network ψ. We find that sampling network ψ significantly improves performance, even without additional data curation steps.

Dataset	CIFAR-10		CIFAR-100		Tiny ImageNet			ImageFruit	ImageMeow	ImageSquawk	ImageYellow
IPC	10	50	10	50	1	10	50	10	10	10	10
NCFM w/o ψ	65.6	74.2	45.9	53.7	9.4	14.2	22.0	39.6	51.6	68.8	67.6
NCFM	**68.9**	**77.4**	**48.7**	**54.4**	**12.0**	**24.3**	**26.5**	**41.4**	**54.4**	**70.8**	**69.2**

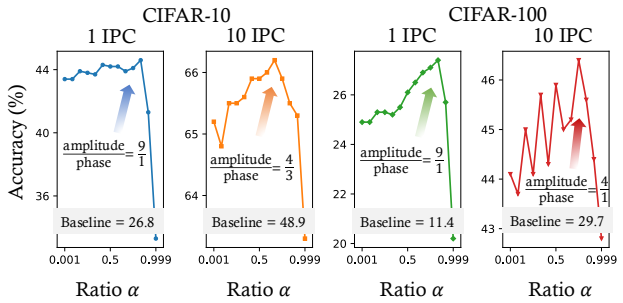

Figure 5. Impact of amplitude and phase components in the NCFD measure across various datasets and IPC settings. The figure illustrates the relationship between the amplitude-to-phase ratio α in Eq. (8). Results indicate that balancing amplitude (for diversity) and phase (for realism) information leads to improved performance. Baseline results were obtained using DM [55].

5.3.3. Effect of the Number of Sampled Frequency Arguments in NCFD

To assess the impact of the number of sampled frequency arguments, $\boldsymbol{t}$, generated by the sampling network ψ, we varied the sample count and measured the corresponding performance. As illustrated in Figure 6, increasing the number of sampled arguments initially enhances the quality of synthetic data by facilitating finer distributional alignment. For example, accuracy on CIFAR-10 at 10 IPC improves from 62% with 16 sampled frequency arguments to approximately 67 % with 1024, indicating a positive correlation between the sampled number and accuracy. However, beyond 1024 arguments, performance gains plateau, with accuracy stabilizing around 67-68% even as the sampling number increases to 4096. This trend suggests that a moderate number achieves an optimal balance between computational efficiency and accuracy. We observed that additional cost remains *minimal* as the number of sampled arguments increases, underscoring NCFM's ability to produce high-quality synthetic data with low computational cost.

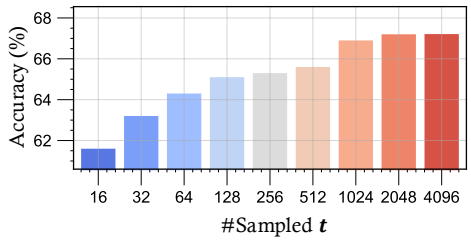

Figure 6. Impact of sampled frequency count in NCFD on accuracy across datasets and IPC. Increasing frequencies improves accuracy up to a threshold, beyond which gains diminish.

6. Discussion

6.1. Training stability of NCFD

The training stability of our minmax paradigm is crucial to its effectiveness. Unlike traditional discrepancy measures, NCFM operates within the complex plane to conduct minmax optimization. While instability is a common issue in minmax adversarial optimization, as seen in generative adversarial networks [2, 37, 39], NCFM consistently maintains stable optimization throughout training, as illustrated in Figure 7. This stability is further supported by theoretical guarantees of weak convergence in Theorem 1, demonstrating the robustness of the CF-based discrepancy under diverse conditions and contributing to NCFM's reliable convergence across datasets.

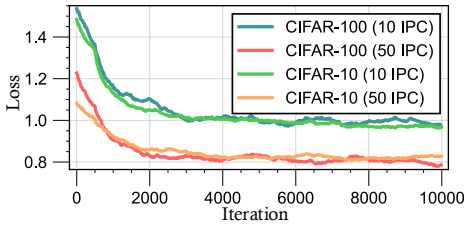

Figure 7. Training dynamics of the minmax optimization process across different datasets and various IPC settings.

6.2. Correlation between CFD and MMD

To better understand NCFM, we examine the relationship between the Characteristic Function Discrepancy (CFD) and Maximum Mean Discrepancy (MMD).

CF as Well-Behaved Kernels in the MMD Metric. The CF discrepancy term $\int_{\boldsymbol{t}} \sqrt{\mathrm{Chf}(\boldsymbol{t}; f)} dF_{\mathcal{T}}(\boldsymbol{t})$ in our loss can be viewed as a well-behaved kernel in MMD, specifically as a *Characteristic Kernel* [43]. Unlike MMD, which relies on fixed kernels, NCFM adaptively learns $F_{\mathcal{T}}(\boldsymbol{t})$, enabling flexible kernel selection for optimal distribution alignment. Furthermore, mixtures of Gaussian distributions within the CF framework produce well-defined characteristic kernels. When MMD employs a characteristic kernel of the form $\int_{\boldsymbol{t}} e^{-j\langle \boldsymbol{t}, \boldsymbol{x}-\tilde{\boldsymbol{x}}\rangle} dF_{\mathcal{T}}(\boldsymbol{t})$, it aligns with the structure of CFD, demonstrating that *MMD is a special case of CFD* when only specific moments are matched. This insight also explains the minimal memory overhead observed as IPC grows, highlighting the efficiency of our approach.

Computational Advantage of CFD over MMD. In contrast to MMD, which requires *quadratic* time in the number

of samples for approximate computation, CFD operates in *linear* time relative to the sampling number of frequency arguments, which aligns results in [1]. This efficiency makes CFD substantially faster and more scalable than MMD, offering a particular advantage for large-scale datasets.

7. Conclusion

In this work, we redefined distribution matching for dataset distillation as a minmax optimization problem and introduced Neural Characteristic Function Discrepancy (NCFD), a novel and theoretically grounded metric designed to maximize the separability between real and synthetic data. Leveraging the Characteristic Function (CF), our method dynamically adjusts NCFD to align both phase and amplitude information in the complex plane, achieving a balance between realism and diversity. Extensive experiments demonstrated the computational efficiency of our approach, which achieves state-of-the-art performance with minimal computational overhead, showcasing its scalability and practicality for large-scale applications.

References

[1] Abdul Fatir Ansari, Jonathan Scarlett, and Harold Soh. A characteristic function approach to deep implicit generative modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7478–7487, 2020. 3, 5, 9

[2] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017. 8

[3] Patrick Billingsley. *Probability and measure*. John Wiley & Sons, 2017. 2, 3

[4] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 2

[5] Torben Maack Bisgaard and Zoltán Sasvári. *Characteristic functions and moment sequences: positive definiteness in probability*. Nova Publishers, 2000. 2, 3

[6] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4750–4759, 2022. 1, 2, 3, 5, 6

[7] Dingfan Chen, Raouf Kerkouche, and Mario Fritz. Private set generation with discriminative information. *Advances in Neural Information Processing Systems*, 35:14678–14690, 2022. 1

[8] Ming-Yu Chung, Sheng-Yen Chou, Chia-Mu Yu, Pin-Yu Chen, Sy-Yen Kuo, and Tsung-Yi Ho. Rethinking backdoor attacks on dataset distillation: A kernel method perspective. *arXiv preprint arXiv:2311.16646*, 2023. 1

[9] Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Scaling up dataset distillation to imagenet-1k with constant memory. In *International Conference on Machine Learning*, pages 6565–6590. PMLR, 2023. 3, 5, 6

[10] Wenxiao Deng, Wenbin Li, Tianyu Ding, Lei Wang, Hongguang Zhang, Kuihua Huang, Jing Huo, and Yang Gao. Exploiting inter-sample and inter-feature relations in dataset distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17057–17066, 2024. 2, 3, 5

[11] Tian Dong, Bo Zhao, and Lingjuan Lyu. Privacy for free: How does dataset condensation help privacy? In *International Conference on Machine Learning*, pages 5378–5396. PMLR, 2022. 1

[12] Jiawei Du, Yidi Jiang, Vincent YF Tan, Joey Tianyi Zhou, and Haizhou Li. Minimizing the accumulated trajectory error to improve dataset distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3749–3758, 2023. 3, 5, 6

[13] Leonard Euler. On transcending quantities arising from the circle. *Chapter*, 8, 1748. 2

[14] Andrey Feuerverger and Roman A Mureika. The empirical characteristic function and its applications. *The annals of Statistics*, pages 88–97, 1977. 2, 4

[15] Jianyang Gu, Kai Wang, Wei Jiang, and Yang You. Summarizing stream data for memory-constrained online continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12217–12225, 2024. 1

[16] Ziyao Guo, Kai Wang, George Cazenavette, Hui Li, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching. *arXiv preprint arXiv:2310.05773*, 2023. 3, 5, 6

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[18] Jeremy Howard and Sylvain Gugger. Fastai: a layered api for deep learning. *Information*, 11(2):108, 2020. 6

[19] Seoungyoon Kang, Youngsun Lim, and Hyunjung Shim. Label-augmented dataset distillation. *arXiv preprint arXiv:2409.16239*, 2024. 5

[20] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. In *International Conference on Machine Learning*, pages 11102–11118. PMLR, 2022. 2, 3, 6

[21] Stephen M Kogon and Douglas B Williams. Characteristic function based estimation of stable distribution parameters. *A practical guide to heavy tails: statistical techniques and applications*, pages 311–338, 1998. 2

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5, 7

[23] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 5

[24] Saehyung Lee, Sanghyuk Chun, Sangwon Jung, Sangdoo Yun, and Sungroh Yoon. Dataset condensation with contrastive signals. In *International Conference on Machine Learning*, pages 12352–12364. PMLR, 2022. 2, 3, 5

[25] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. *Advances in neural information processing systems*, 30, 2017. 2

[26] Hongcheng Li, Yucan Zhou, Xiaoyan Gu, Bo Li, and Weiping Wang. Diversified semantic distribution matching for dataset distillation. In *ACM Multimedia 2024*, 2024. 5

[27] Shengxi Li, Zeyang Yu, Min Xiang, and Danilo Mandic. Reciprocal adversarial learning via characteristic functions. *Advances in Neural Information Processing Systems*, 33:217–228, 2020. 3, 5

[28] Shengxi Li, Jialu Zhang, Yifei Li, Mai Xu, Xin Deng, and Li Li. Neural characteristic function learning for conditional image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7204–7214, 2023. 5

[29] Zhe Li and Bernhard Kainz. Image distillation for safe data sharing in histopathology. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 459–469. Springer, 2024. 1

[30] Dai Liu, Jindong Gu, Hu Cao, Carsten Trinitis, and Martin Schulz. Dataset distillation by automatic training trajectories. *arXiv preprint arXiv:2407.14245*, 2024. 5

[31] Paul Lévy. *Théorie de l'addition des variables aléatoires*. Gauthier-Villars, Paris, 1937. 2, 4, 5

[32] Danilo P Mandic and Anthony G Constantinides. Complex valued nonlinear adaptive filters: state of the art. *Signal Processing*, 89(9):1704–1725, 2009. 5, 7

[33] Dmitry Medvedev and Alexander D'yakonov. Learning to generate synthetic training data using gradient matching and implicit differentiation. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 138–150. Springer, 2021. 1

[34] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. *arXiv preprint arXiv:2011.00050*, 2020. 5

[35] Alan V Oppenheim and Jae S Lim. The importance of phase in signals. *Proceedings of the IEEE*, 69(5):529–541, 1981. 5, 7

[36] Tian Qin, Zhiwei Deng, and David Alvarez-Melis. A label is worth a thousand images in dataset distillation. *arXiv preprint arXiv:2406.10485*, 2024. 5

[37] Alec Radford. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 8

[38] Ahmad Sajedi, Samir Khaki, Ehsan Amjadian, Lucy Z Liu, Yuri A Lawryshyn, and Konstantinos N Plataniotis. Datadam: Efficient dataset distillation with attention matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17097–17107, 2023. 2, 3

[39] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 8

[40] Shitong Shao, Zeyuan Yin, Muxin Zhou, Xindong Zhang, and Zhiqiang Shen. Generalized large-scale data condensation via various backbone and statistical matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16709–16718, 2024. 5

[41] Neil G Shephard. From characteristic function to distribution function: a simple framework for the theory. *Econometric theory*, 7(4):519–529, 1991. 2

[42] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[43] Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. *The Journal of Machine Learning Research*, 11:1517–1561, 2010. 8

[44] Felipe Petroski Such, Aditya Rawal, Joel Lehman, Kenneth Stanley, and Jeffrey Clune. Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data. In *International Conference on Machine Learning*, pages 9206–9216. PMLR, 2020. 1

[45] Peng Sun, Bei Shi, Daiwei Yu, and Tao Lin. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9390–9399, 2024. 5, 6

[46] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018. 5

[47] Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. Cafe: Learning to condense dataset by aligning features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12196–12205, 2022. 1, 2, 3, 5, 6

[48] Shaobo Wang, Yantai Yang, Shuaiyu Zhang, Chenghao Sun, Weiya Li, Xuming Hu, and Linfeng Zhang. Drupi: Dataset reduction using privileged information, 2024. 6

[49] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. 1, 3

[50] Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th annual international conference on machine learning*, pages 1121–1128, 2009. 5

[51] Enneng Yang, Li Shen, Zhenyi Wang, Tongliang Liu, and Guibing Guo. An efficient dataset condensation plugin and its application to continual learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1

[52] Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1

[53] Hansong Zhang, Shikun Li, Pengju Wang, Dan Zeng, and Shiming Ge. M3d: Dataset condensation by minimizing maximum mean discrepancy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9314–9322, 2024. 2, 3, 5

[54] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021. 2, 3, 5, 6

[55] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching, 2022. 1, 2, 3, 5, 6, 8

[56] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*, 2020. 1, 2, 3, 5

[57] Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7856–7865, 2023. 2, 3, 5, 6

[58] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. *Advances in Neural Information Processing Systems*, 35:9813–9827, 2022. 5