Author Contributions

In this paper, the authors made the following contributions:

- Jiawei Du developed the theoretical framework, and proposed FTD. He also designed the experiments, analyzed the results, plotted the figures, and wrote the majority of the manuscript.
- Yidi Jiang implemented FTD and conducted the experiments. She recorded the experimental logs and analyzed the results. She also wrote the experimental and related works sections.
- Vincent Y. F. Tan guided the formulation of FTD. He also helped develop the theoretical framework and revised the manuscript.
- Joey Tianyi Zhou and Haizhou Li supervised the project and provided critical feedback on the research.

A. More Discussions and Experiments

A.1. Exploring the Accumulated Trajectory Error

We design experiments on the CIFAR-100 dataset with $\texttt{ipc} = 10$ to verify the existence and observe the adverse effect of the accumulation of the trajectory error (as defined in Equation 8) of MTT.

We present the loss difference $L_{\mathcal{T}_{\text{Test}}}(f_\theta) - L_{\mathcal{T}_{\text{Test}}}(f_{\theta^*})$, which quantifies how well the student trajectory matches the teacher trajectory along the epochs during evaluation phase, in Figure 1. We also present the loss difference during the distillation phase that serves as a baseline. It can be seen that the loss difference of MTT (blue line) in the evaluation phase accumulates as the evaluation progresses, and is much higher than the one in the distillation phase (cyan line). These results demonstrate the existence the accumulation of the trajectory error ϵ_t. Moreover, the loss difference of FTD (purple line) is shown to be much lower than that of MTT (blue line), which suggests that our proposed FTD reduces the accumulated trajectory error ϵ_t effectively.

Table 6. Ablation results of the initialization discrepancy. The start epoch indicates that the n^{th} epoch's set of weights from the teacher trajectories is used to initialize the network. The epochs to train indicates the remaining epochs to train the initialized network (1 epoch = 20 synthetic steps).

Start Epoch	Epochs to train	MTT Accuracy	Our Accuracy
0	50	35.4	37.7
10	40	37.0	39.5
20	30	38.6	41.6
30	20	40.2	43.5
40	10	42.1	44.4
45	5	42.3	46.2

We also design experiments to show the existence of the initialization error $\mathcal{I}_t$ in Equation 7. Recall that this is the dominant factor leading to the accumulation of the trajectory error as shown in Equation 8. We compare the accuracies of several 3-layer ConvNets [12] trained using the same synthetic dataset $\mathcal{S}$ but initialized with different weights. These networks are initialized by the sets of weights in epochs $0, 5, 10, \ldots, 40, 45$ of the teacher trajectories, and are trained until the 50^{th} epoch. Specifically, the network initialized by the sets of weights in epoch 0 serves as the baseline. These weights are equivalent to being initialized from the student trajectories in epochs $5, 10, \ldots, 40, 45$, respectively. Note that training over the 50 epochs of the teacher trajectories is equal to doing the same over 100 iterations of the student trajectories (1 epoch $= 20$ synthetic steps), which is much fewer than the 1000 iterations trained in the evaluation phase. Thus the accuracy is degraded as compared to Table 1. Following the above settings, we evaluate MTT and FTD and report the results in Table 6. It can be seen that the networks initialized by the sets of weights from the teacher trajectories always outperform the baseline. In fact, the fewer epochs used to train, the better the accuracy. The results clearly show the adverse effect of the initialization discrepancy. A more precise initialization (closer to the initialization used in distillation) will have a more significant impact on the final performance. However, FTD is as expected to suppress the initialization error $\mathcal{I}_t$ so that it eventually surpasses the performance of MTT.

A.2. Exploring the Flat Trajectory

We conducted experiments in subsection 4.3 to show that the performance gain of FTD is primarily due to the regularized flat trajectory. Although a DNN trained on the real dataset will generalize better if the training converges to a flat minimum, unfortunately, the benefit of flat minima is no longer valid if we consider the synthetic dataset. We provide some theoretical explanations here.

We denote $\mathcal{D}$ as the natural distribution, $L_{\mathcal{D}}(f_\theta)$ is equivalent to the expected loss over test set. Each sample in the real training dataset $\mathcal{T}$ is drawn i.i.d. from $\mathcal{D}$. For simplicity, we consider Gaussian priors and likelihoods, in which case the posterior is also Gaussian. Hence, we assume that over the parameter space, $\mathscr{P} = \mathcal{N}(\boldsymbol{\mu}_P, \sigma_P^2\mathbf{I})$ is the prior distribution and $\mathcal{W} = \mathcal{N}(\boldsymbol{\mu}_W, \sigma_W^2\mathbf{I})$ is the posterior distribution trained on $\mathcal{T}$, where $\boldsymbol{\mu}_P, \boldsymbol{\mu}_W \in \mathbb{R}^k$ and $\mathbf{I}$ is the $k \times k$ identity matrix. We assume that the matching error $\delta \sim \mathcal{N}(\mathbf{0}, \sigma_\epsilon^2\mathbf{I})$. Pierre et al. [11] states a generalization bound based on the sharpness to theoretically justify the benefit of flat minima derived from the PAC-Bayesian generalization bound [33] as follows. For $n = |\mathcal{T}|$ and with probability at least $1-\delta$, over the choice of the real training set $\mathcal{T}$, the following inequality

holds

$$\mathbb{E}_{\theta\sim\mathcal{W}}\big[L_{\mathcal{D}}(f_\theta)\big] \leq \mathbb{E}_{\theta\sim\mathcal{W}}\big[L_{\mathcal{T}}(f_\theta)\big] + \Delta L(\mathscr{P}), \quad (14)$$

where

$$\Delta L(\mathscr{P}) = \sqrt{\frac{\mathrm{KL}(\mathcal{W}\|\mathscr{P}) + \log\frac{n}{\delta}}{2(n-1)}}.$$

In this bound ΔL quantifies the generalization error, i.e., the closeness between the test and training losses. As we stated in subsection 3.3, the gradient-matching dataset distillation is equivalent to mapping a initialization distribution P_{θ_0} into the posterior distribution $\mathcal{W}$. However, due to the existence of the matching error, the posterior distribution $\tilde{\mathcal{W}}$ trained on the synthetic set $\mathcal{S}$ is more dispersed than $\mathcal{W}$, i.e., $\tilde{\mathcal{W}} = \mathcal{N}(\boldsymbol{\mu}_W, \sigma_W^2\mathbf{I} + \sigma_\epsilon^2\mathbf{I})$ for some $\sigma_\epsilon^2 \geq 0$. Since the KL divergence can be written as [11],

$$\begin{aligned}
&\mathrm{KL}(\mathcal{W}\|\mathscr{P}) \\
&= \frac{1}{2}\left[\frac{k\sigma_W^2 + \|\boldsymbol{\mu}_P - \boldsymbol{\mu}_W\|_2^2}{\sigma_P^2} - k + k\log\left(\frac{\sigma_P^2}{\sigma_W^2}\right)\right] \\
&= \frac{k}{2}\left[\frac{\sigma_W^2}{\sigma_P^2} - \log\frac{\sigma_W^2}{\sigma_P^2}\right] + \frac{1}{2}\left[\frac{\|\boldsymbol{\mu}_P - \boldsymbol{\mu}_W\|_2^2}{\sigma_P^2} - k\right],
\end{aligned}$$

where k is the number of parameters. Therefore, we have

$$\begin{aligned}
&\mathrm{KL}(\tilde{\mathcal{W}}\|\mathscr{P}) - \mathrm{KL}(\mathcal{W}\|\mathscr{P}) \\
&= \frac{k}{2}\left[\left(\frac{\sigma_W^2 + \sigma_\epsilon^2}{\sigma_P^2} - \log\frac{\sigma_W^2 + \sigma_\epsilon^2}{\sigma_P^2}\right) - \left(\frac{\sigma_W^2}{\sigma_P^2} - \log\frac{\sigma_W^2}{\sigma_P^2}\right)\right] \\
&\geq 0.
\end{aligned}$$

The final inequality holds as $\sigma_\epsilon^2 \geq 0$ and $\sigma_W^2 \geq \sigma_P^2$. Consequently, the generalization error $\Delta L(\tilde{\mathcal{W}})$ over the synthetic dataset $\mathcal{S}$ will be greater than $\Delta L(\mathcal{W})$ over the real dataset $\mathcal{T}$. The experiments in subsection 4.3 verify that the flat minima of the synthetic dataset does not benefit generalization ability as the generalization bound in Equation 14 is loose.

A.3. Implementation Details

A.3.1 Parameter Study

The coefficient ρ in Equation 12 controls the amplitude of the perturbation ϵ, which affects the flatness of the obtained teacher trajectories [11]. We study the effect of ρ by using grid searches from the set $\{0.005, 0.01, 0.03, 0.05, 0.1\}$ during the buffer phase. We report the accuracies of the evaluated synthetic dataset in Figure 5. We observe that $\rho = 0.01$ achieves the best improvement , which is different from the suggested value $\rho = 0.05$ [11]. Lastly, it is not sensitive to choose the value of ρ as FTD outperforms MTT with every evaluated value of ρ.

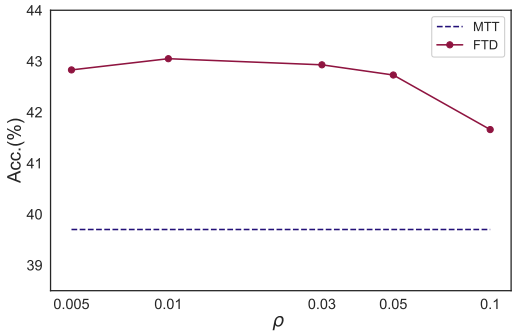

Figure 5. Parameter study of ρ on CIFAR-100 (ipc=10). We set the x-axis to be in log scale for better illustration. Blue dashed line is the result of MTT, which serves as the baseline.

A.3.2 Optimizing of the Flat Trajectory

As introduced in subsection 3.3, FTD only regularizes the training in the buffer phase as in Equation 13 to obtain a flat teacher trajectory. We provide the pseudocode for reproducing our results in Algorithm 1. The optimization of the flat trajectory is solving a minimax problem. We follow Pierre et al. [11] to approximate the solution $\hat{\epsilon}$ of the maximization in Equation 12 as follows

$$\begin{aligned}
\hat{\epsilon} &= \arg\max_{\epsilon\in\Psi} \big[L_{\mathcal{T}}(f_{\theta+\epsilon}) - L_{\mathcal{T}}(f_\theta)\big] \\
&= \rho\frac{\nabla_\theta L_{\mathcal{T}}(f_\theta)}{\|\nabla_\theta L_{\mathcal{T}}(f_\theta)\|_2},
\end{aligned} \quad (15)$$

where $\Psi = \{\epsilon : \|\epsilon\|_2 \leq \rho\}$ and $\rho > 0$ is a given constant that determines the permissible norm of ϵ. We denote $g_L = \nabla_\theta L_{\mathcal{T}}(f_{\theta_t})$, which is the gradient to optimize the vanilla loss function $L_{\mathcal{T}}(f_\theta)$. Hence, from Equation 15, we have that

$$\hat{\epsilon} = \rho\frac{g_L}{\|g_L\|_2}.$$

Suppose that $\theta^{\text{adv}} = \theta + \hat{\epsilon}$, we can rewrite Equation 13 as follows,

$$\begin{aligned}
\theta^* &= \arg\min_\theta \big\{L_{\mathcal{T}}(f_\theta) + \alpha\, S(\theta)\big\} \\
&= \arg\min_\theta \big\{L_{\mathcal{T}}(f_\theta) + \alpha\,[L_{\mathcal{T}}(f_{\theta^{\text{adv}}}) - L_{\mathcal{T}}(f_\theta)]\big\} \\
&= \arg\min_\theta \big\{\alpha\, L_{\mathcal{T}}(f_{\theta^{\text{adv}}}) + (1-\alpha)[L_{\mathcal{T}}(f_\theta)]\big\}.
\end{aligned} \quad (16)$$

We denote $g_{S+L} = \nabla_\theta L_{\mathcal{T}}(f_{\theta^{\text{adv}}})$, which is the gradient to optimize $L_{\mathcal{T}}(f_{\theta^{\text{adv}}})$. Hence, from Equation 16, the gradient to optimize θ^* is $g = \alpha\cdot g_{S+L} + (1-\alpha)\cdot g_L$ as illustrated in Line 6 of Algorithm 1. The parameter α is found using a grid search, as described next.

Algorithm 1 Training with FTD in buffer phase.

Input: Real set $\mathcal{T}$; A network f with weights θ; Learning rate η; Epochs E; Iterations T per epoch; FTD hyperparameter α, ρ.

for $e = 1$ to E **do**
  for $t = 1$ to T, Sample a mini-batch $\mathcal{B} \subset \mathcal{T}$ **do**
    Compute gradients $g_L = \nabla_\theta L_\mathcal{B}(f_{\theta_t})$
    $\theta_t^{\text{adv}} = \theta_t + \rho \cdot \frac{g_L}{\|g_L\|_2}$
    Compute gradients $g_{S+L} = \nabla_\theta L_\mathcal{B}(f_{\theta_t^{\text{adv}}})$
    Compute $g = \alpha \cdot g_{S+L} + (1-\alpha) \cdot g_L$
    Update weights $\theta_{t+1} \leftarrow \theta_t - \eta g$
  Record weights θ_T ▷ Record the trajectory at the end of each epoch

Output: A flat teacher trajectory.

A.3.3 Hyperparameter Details

The hyperparameters α and ρ of FTD are obtained via grid searches in a validation set within the CIFAR-10 dataset. The hyperparameter ρ is searched within the set $\{0.005, 0.01, 0.03, 0.05, 0.1\}$. The hyperparameter α is searched within the set $\{0.1, 0.3, 0.5, 1.0, 3.0\}$. For the rest of the hyperparamters, we report them in Table 7.

A.3.4 Neural Architecture Search.

Following the search space construction of 720 ConvNets in [50], we vary the different hyperparameters including the width $W \in \{32, 64, 128, 256\}$, depth $D \in \{1, 2, 3, 4\}$, normalization $N \in \{\text{None, BatchNorm, LayerNorm, InstanceNorm, GroupNorm}\}$, activation $A \in \{\text{Sigmoid, ReLU, LeakyReLU}\}$, pooling $P \in \{\text{None, MaxPooling, AvgPooling}\}$. Every candidate ConvNet is trained with the proxy dataset, and then evaluated on the whole test dataset. These candidate ConvNets are then ranked by their test performances. The architectures with the top 5, 10 and 20 test accuracies are selected and the Spearman's rank correlation coefficients between the searched rankings of the synthetic dataset and the real dataset are computed after training. We train each ConvNet for a total of 3 times to obtain averaged validation and test accuracies.

A.3.5 Visualizations

We provide more visualizations of the synthetic datasets for `ipc` $= 1$ from the different resolution datasets: 32×32 CIFAR-10 dataset in Figure 6, 64×64 Tiny ImageNet dataset in Figure 7, 128×128 ImageNette subset in Figure 8. In addition, parts of the visualizations of synthetic images from the CIFAR-100 dataset are showed in Figure 9.

B. More Related Work

Dataset Distillation. Dataset distillation presented by [45] aims to obtain a new, synthetic dataset that is much reduced in size which also performs almost as well as the original dataset. Similar to [45], several approaches consider end-to-end training [34, 36], however they frequently necessitate enormous computation and memory resources and suffer from inexact relaxations [34, 36] or training instabilities caused by unrolling numerous iterations [32, 45]. Other strategies [48, 50] lessen the difficulty of optimization by emphasizing short-term behavior, requiring a single training step on the distilled data to match that on the real data. Nevertheless, errors may accrue during evaluation, when the distilled data is used in multiple steps.

To address the difficulties of error accumulation in single training step matching algorithms [48, 50], Cazenavette et al. [1] propose to match segments of the parameter trajectories trained on synthetic data with long-range training trajectory segments of networks trained on the real datasets. However, the error accumulation of the parameters in particular segments is still inevitable. Instead, our strategy further mitigates the accumulated trajectory errors with the guidance of a flat teacher trajectory inspired by the heuristic of Sharpness-aware Minimization.

The geometry of the loss landscape. Minimizing the spectrum of the Hessian matrix $\nabla_\theta^2 f_\theta$ as in Equation 11 is an difficult and expensive task. Fortunately, a series of sharpness-aware minimization methods [10, 11, 51] have been proposed to perform the task implicitly with low cost for improved generalization. It has been argued in many studies [7, 17, 18, 20] that the spectrum of the Hessian matrix constitutes a good characterization of the geometry of the loss landscape (sharpness), which then translates to having a strong relationship to the generalization abilities [7, 30, 31] of the network. We leverage the approaches from [11, 51] to efficiently optimizing the spectrum of the Hessian matrix to minimize the accumulated trajectory error in this work.

Table 7. Hyperparameter values we used for the main result table.

	CIFAR-10			CIFAR-100			Tiny ImageNet	
ipc	1	10	50	1	10	50	1	10
Synthetic Step	50	30	30	40	20	80	30	20
Expert Epoch	2	2	2	3	2	2	2	2
Max Start Epoch	2	20	40	20	40	40	10	40
Synthetic Batch Size	-	-	-	-	-	1000	-	500
Learning Rate (Pixels)	100	100	1000	1000	1000	1000	10000	10000
Learning Rate (Step Size)	1e-7	1e-5	1e-5	1e-5	1e-5	1e-5	1e-4	1e-4
Learning Rate (Teacher)	0.01	0.001	0.01	0.01	0.01	0.01	0.01	0.01
α	0.3	0.3	1	1	1	1	1	1
EMA Decay	0.9999	0.9995	0.999	0.9995	0.9995	0.999	0.999	0.999

Figure 6. Visualizations of synthetic images distilled from the 32×32 CIFAR-10 dataset with $\texttt{ipc} = 1$.

Figure 7. Visualizations of part of synthetic images distilled from the 64×64 Tiny ImageNet dataset with $\texttt{ipc} = 1$.

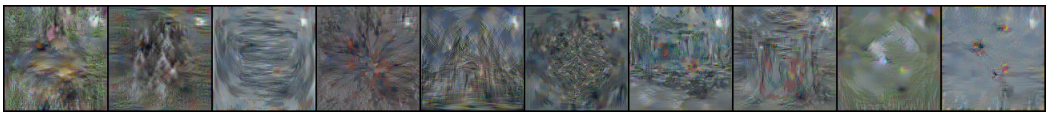

Figure 8. Visualizations of synthetic images distilled from the 128×128 ImageNette subset with $\texttt{ipc} = 1$.

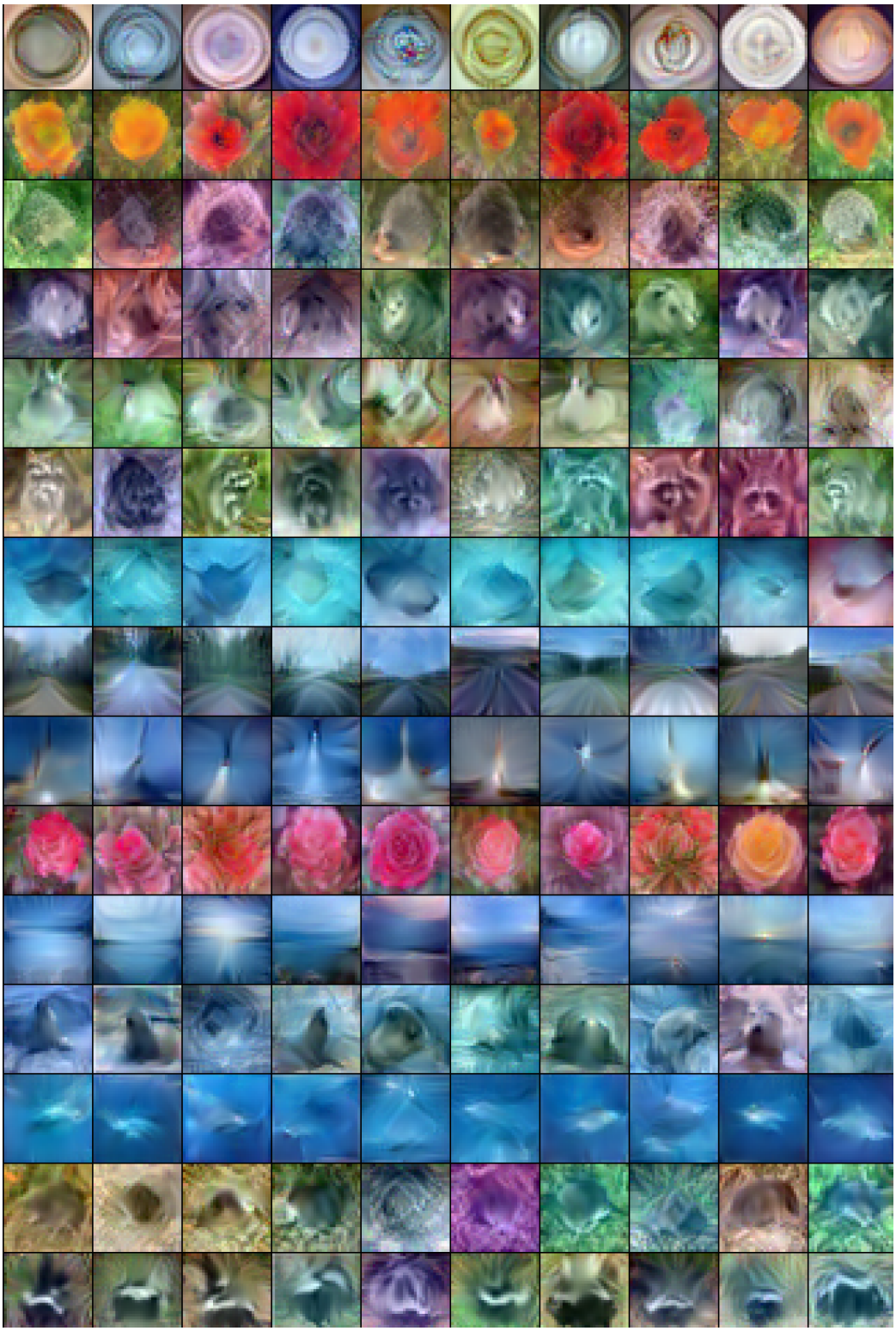

Figure 9. Visualizations of part of synthetic images distilled from the 32×32 CIFAR-100 dataset with $\texttt{ipc} = 10$.