Dataset Distillation by Influence Matching

Haoru Tan[1,†] Wang Wang[1,†] Sitong Wu[2] Xiuzhe Wu[3]

Yang-Tian Sun[1] Chirui Chang[1] Shaofeng Zhang Xiaojuan Qi[1,✉]

[1]HKU [2]CUHK [3]Stanford

hrtan@eee.hku.hk

Abstract

We revisit dataset distillation from an outcome-centric perspective. Rather than aligning process surrogates (per-step gradients or training trajectories), Influence Matching (Inf-Match) aligns the final outcome of training: it learns a compact synthetic set whose effect on the converged parameters matches that of the full dataset. Concretely, we introduce a fully differentiable, sample-level influence estimator that quantifies parameter shifts from adding or removing data, without time-consuming inverse-Hessian products or convexity assumptions. The estimator runs in linear time by unrolling the optimization dynamics and applying a first-order Taylor approximation. We then learn the synthetic set by minimizing the mismatch between its influence and that of the real dataset, yielding outcome alignment rather than heuristic process imitation. Inf-Match delivers the best accuracy across standard classification benchmarks. For instance, on Tiny-ImageNet (IPC=10), Inf-Match attains 31.5%, a +4.7% improvement over NCFM. Beyond classification, Inf-Match scales to vision-language distillation on Flickr30K, outperforming strong process-matching baselines. For instance, with 200 to 1000 synthetic samples, our method achieved a leading impressive average on image/text retrieval tasks, higher than NCFM by 2.5%. The code will be released via https://github.com/hrtan/infmatch.

1. Introduction

With the rapid expansion of visual data across domains such as autonomous driving, surveillance, and web-scale imagery, the size of modern datasets has grown to tens or even hundreds of millions of samples. While large-scale data fuels the success of deep vision models, it also brings prohibitive costs in storage, transmission, and training. These challenges have motivated dataset distillation, the task of synthesizing a small but highly informative dataset that encapsulates the knowledge of a much larger one [33, 55, 66, 67]. By learning a handful of representative samples that preserve the training dynamics or final performance of the full dataset, dataset distillation enables efficient data sharing [66, 68], rapid model adaptation, privacy-preserved learning [17], and continual or federated learning [23, 61] under strict resource budgets.

Early efforts tackle this problem by approximating the original bilevel optimization of dataset distillation (See Eq. (1)) with more tractable proxies, like feature matching [52, 66, 67] and (optimization) process matching [7, 68]. The latter has received more attention due to its generally superior performance. Typical process matching approaches include Gradient Matching (GM) [68], which aligns per-step gradients between real and synthetic data, and Trajectory Matching (MTT) [7], which enforces consistency between training trajectories. These methods have achieved encouraging progress, yet they rely on heuristic surrogates that only mimic intermediate training behaviors. Consequently, the synthetic data may perfectly reproduce the training process of the real data without guaranteeing comparable performance or generalization.

Despite recent advancements such as improved feature distribution matching [54] or difficulty-aware trajectory alignment [24], a fundamental limitation persists: these alignment proxies do not imply outcome alignment. The ultimate goal of dataset distillation is not to match training steps but to reproduce the influence that the full dataset exerts on the learned model. Bridging this *optimization gap* demands a principled way to quantify how individual samples or subsets influence the final trained model. However, existing influence estimators are computationally prohibitive [4, 22, 28, 29, 31, 43], requiring inverse-Hessian computations and assuming convex losses that fail for deep neural

† Equal first author.

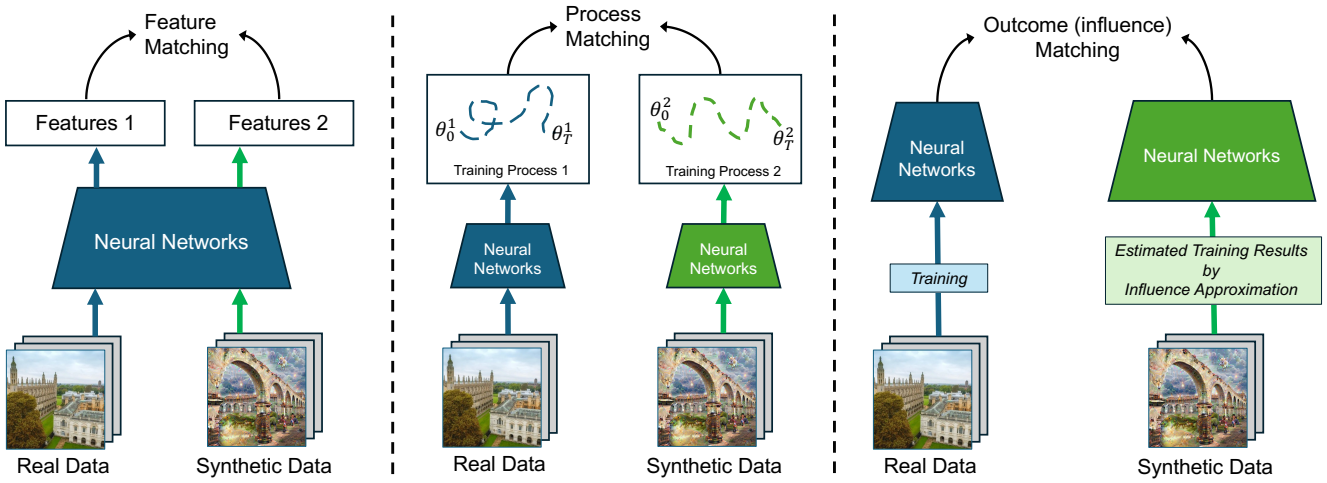


Figure 1. This figure outlines three core paradigms for data comparison or generation: (a) Feature matching, where a feature extractor is trained on real data, and synthetic data is generated to match the extracted features [54, 66, 67]; (b) (Optimization) Process matching, which seeks to align the optimization paths [7, 24] or the gradients [68] of models trained on synthetic and real data; and (c) Our outcome matching pipeline, which focuses on matching the final trained models (outcome) resulting from both synthetic and real data. Crucially, the model trained on synthetic data in our methodology is not obtained through actual retraining; rather, its parameters are effectively estimated by leveraging our novel influence estimator, see Remark 1 for details.

networks. As a result, a key open question remains: can we directly distill data by matching its outcome influence on the final model, rather than its intermediate dynamics?

We address this question by introducing **Influence Matching (Inf-Match)**, a new dataset distillation method that shifts the focus from process alignment to *outcome alignment*. The core of our framework is a *differentiable sample influence estimator* that quantifies how a data sample or a group of samples contributes to the final optimized model parameters without any convexity assumptions or inverse-Hessian computation. without convexity assumptions and without inverse–Hessian computations. We derive this estimator by unrolling the optimization dynamics and applying a first-order Taylor approximation, yielding *linear-time* complexity and high-fidelity estimates that are practical for scalable real-world applications. Then, **Inf-Match** optimizes the synthetic dataset so that its influence on the model matches the influence of the original dataset, effectively ensuring that training on the synthetic data yields the same final model as training on the full data. This formulation markedly narrows the optimization gap in dataset distillation and establishes a direct path toward outcome-aligned distillation.

Extensive experiments validate the effectiveness and generality of our approach. On CIFAR-10, CIFAR-100, and Tiny-ImageNet, **Inf-Match** consistently achieves the best performance, surpassing strong baselines such as DATM [24] and NCFM [54] across all image-per-class (IPC) settings, see Figure 2. For instance, on Tiny-ImageNet (IPC=10), **Inf-Match** attains 31.5%, a +4.7% improvement over NCFM. It further demonstrates strong scalability to vision-language datasets such as Flickr30K. For instance, with 200 synthetic samples, our method achieved an impressive score on image-to-text retrieval tasks that is higher than the next best, DATM [24], at 1.3%. Moreover, with 200 to 1000 synthetic samples, our method achieved a leading impressive average on image/text retrieval tasks, higher than NCFM by 2.5%, see Figure 4.

2. Related Works

Dataset Distillation Given the burdensome nature of the original problem in Eq. (1), one has to explore proxy tasks, such as feature matching [42, 65, 66, 69] and (optimization) process matching [7, 13, 19–21, 24, 67]. In the following, we mainly review these works.

(1). Feature matching. There is one line of work [42, 65, 66, 69] that tries to match the latent feature space directly. Distribution matching (DM) [66] proposed to match the synthetic and target data from the distribution perspective for dataset distillation. CAFE [52] improved the distribution matching from several aspects: (1) using multiple-layer features other than only the last-layer features for matching, (2) proposing the discrimination loss to enlarge the class distinction of synthetic data. IDM [69] adds a classification loss as regularization to mitigate less classified synthetic data caused by the first-order moment mean matching. Datadam [42] proposed to learn synthetic images by matching the spatial attention maps of real and synthetic data generated by different layers within a family of randomly initialized

neural networks. M3D [65] proposed to minimize the maximum mean discrepancy (MMD) between the real and the synthetic data.

(2). Process matching. Another group of methods [7, 13, 19, 20, 24, 67, 68] constructs the surrogate problem of matching the intermediate training state contributed by the synthetic data and the real data, respectively. Among them, the most representative schemes are matching gradient [68] in training and matching trajectory [7] in training. DSA [67] proposed incorporating the gradient matching framework with a differentiable augmentation scheme to synthesize more informative synthetic images and for better performance when training networks with augmentations. SeqMatch [20] addresses the issue of failing to condense high-level features in dataset distillation. It divides synthetic data into multiple subsets, sequentially optimizing them to promote the effective distillation of high-level features learned in later epochs. MTT [7] proposes a new formulation that optimizes our distilled data to guide networks to a similar state as those trained on real data across many training steps. DATM [24] also distills easy/difficult information into the trajectory matching framework, achieving further performance improvements.

(2). Others. Beyond the above, some recent studies [48, 63] have integrated considerations of data diversity and the authenticity of synthesized data into their framework designs. Additionally, research [8, 53, 64] has investigated the applicability of generative models in dataset distillation. Furthermore, both meta-gradient-based methods [15, 47, 55] and kernel-based methods [36, 38, 71] have been explored in dataset distillation. It is worth noting that Kernel-based methods [36, 38, 71] can theoretically estimate the results of inner-loop training directly, avoiding the need for inner-loop training. However, a significant challenge is that by approximating the learning process with a linear kernel, the kernel may overlook the complex training dynamics [3]. This, in turn, results in a decline in overall performance.

Influence Estimation. Influence estimation [9, 12, 25, 29, 45, 49–51] focuses on linking the training data to the performance of a model post-training. A common approach to assessing the impact of a specific data point is through leave-one-out (LOO) retraining. This technique involves training the model on the dataset while omitting certain samples, then evaluating the changes in performance compared to the model trained with the complete dataset. Instead of relying on full retraining, Koh et al. [29] suggest an alternative method that estimates the effects on the model from slight modifications in the weights of the training data. Their approach utilizes an estimator derived from the product of the inverse Hessian and the gradient, allowing for an efficient approximation of the influence exerted by individual samples. In subsequent research, various initiatives have sought to enhance this method along different avenues. To improve scalability, several approaches have been introduced [22, 31, 43]. Regarding the estimation precision, some studies have focused on analyzing [28] and refining [4] the influence function to better assess group impacts.

However, these works have some significant shortcomings. Firstly, they depend on the rather stringent assumption that the loss function is convex to parameters, a condition that is frequently not met [11, 14]. Secondly, scaling these methods to accommodate large models and extensive datasets poses challenges, primarily due to the computational burden associated with the inverse-Hessian-gradient product. These factors significantly restrict their applicability.

3. Preliminaries

Given a real dataset $\mathcal{D}$, a deep network with parameters $\boldsymbol{\theta}$, and a task loss $\mathcal{L}(\cdot,\cdot)$, *dataset distillation* seeks a compact synthetic set $\mathcal{S}$ such that training on $\mathcal{S}$ yields a model that performs comparably to one trained on $\mathcal{D}$. This is naturally posed as a bilevel program [55, 68]:

$$\underbrace{\mathcal{S}^* = \arg\min_{\mathcal{S}} \mathcal{L}(\mathcal{D}, \boldsymbol{\theta}^*_{\mathcal{S}})}_{\text{Outer-Level: Data Optimization}} \quad \text{s.t.} \quad \underbrace{\boldsymbol{\theta}^*_{\mathcal{S}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{S}, \boldsymbol{\theta})}_{\text{Inner-Level: Network Optimization}} , \tag{1}$$

where the inner problem trains the network on $\mathcal{S}$ to obtain $\boldsymbol{\theta}^*_{\mathcal{S}}$, and the outer problem updates $\mathcal{S}$ so that the resulting model minimizes loss on $\mathcal{D}$. In effect, the distilled set $\mathcal{S}$ is learned to induce a final model whose performance closely mirrors that of training on the full dataset $\mathcal{D}$.

Dataset distillation is, by nature, a bi-level optimization problem, which is notoriously hard to solve in practice. Consequently, many methods resort to proxy objectives [24, 66, 68], such as trajectory matching [7] or gradient alignment [68]. These proxies rest on a heuristic assumption: if the synthetic data can mimic intermediate training signals (e.g., gradients, parameter states), it will yield the same outcome as the full dataset. This assumption is fragile. It induces an optimization gap in which synthetic data can excel at the proxy objective, yet still underperform on downstream accuracy and generalization. In effect, there is a fundamental trade-off between computational tractability (via process alignment) and fidelity to the true objective (matching the final, outcome-level behavior of the model).

4. Method

Here, we present the formulation of our proposed dataset distillation framework, **Influence Matching (Inf-Match)**. We begin by defining the influence of individual samples or subsets of data on the final trained model and then introduce our efficient estimator.

4.1. Data Influence

Given a model with parameters $\boldsymbol{\theta}$ to be trained, and an original training dataset $\mathcal{D}$, the **removal** influence of a single sample or a group of samples $\mathcal{Z} \subset \mathcal{D}$, denoted as $\mathcal{I}_{-\mathcal{Z}}$, is defined by the difference in the model's final parameters when trained without and with $\mathcal{Z}$:

$$\textbf{Removal-Influence:} \quad \mathcal{I}_{-\mathcal{Z}} = \boldsymbol{\theta}^*_{\mathcal{D}-\mathcal{Z}} - \boldsymbol{\theta}^*_{\mathcal{D}},$$

where $\boldsymbol{\theta}^*_{\mathcal{D}}$ represents the optimal parameters of the model trained on the full dataset $\mathcal{D}$, and $\boldsymbol{\theta}^*_{\mathcal{D}-\mathcal{Z}}$ represents the optimal parameters of the model trained on the dataset with the set $\mathcal{Z}$ removed. Intuitively, $\mathcal{I}_{-\mathcal{Z}}$ quantifies the exact change in the model's final state that is attributable to the presence of the data $\mathcal{Z}$ during training. Analogously, we define the influence resulting from the **addition** of a set of external samples $\mathcal{Z} \not\subset \mathcal{D}$ (which originally do not belong to the training set $\mathcal{D}$) as:

$$\textbf{Addition-Influence:} \quad \mathcal{I}_{+\mathcal{Z}} = \boldsymbol{\theta}^*_{\mathcal{D}+\mathcal{Z}} - \boldsymbol{\theta}^*_{\mathcal{D}},$$

where $\boldsymbol{\theta}^*_{\mathcal{D}+\mathcal{Z}}$ represents the optimal parameters of the model trained on the combined dataset $\mathcal{D} \cup \mathcal{Z}$. This quantity $\mathcal{I}_{+\mathcal{Z}}$ captures the exact shift in the final model parameters caused by introducing the new data $\mathcal{Z}$.

Data Influence Estimator. Existing influence estimators [22, 28, 29, 31, 43] suffer from two major limitations: (i) they rely on the convexity of the loss landscape, which rarely holds for deep networks, and (ii) they require computing inverse–Hessian products, leading to high computational overhead [29, 31]. To overcome these issues, we develop an efficient, fully differentiable estimator by unrolling the optimization dynamics and applying a first-order Taylor approximation. It achieves *linear-time* complexity, avoids any inverse–Hessian computation by Theorem 1, and is supported by a provable (tight) upper bound on the estimation error, see Theorem (2) for more details.

Theorem 1 ***(Influence estimator)*** *Let $\{(\boldsymbol{\theta}^t_{\mathcal{D}}, \eta_t)|^T_{t=1}\}$ denote a series of parameters and learning rates used during the model training on $\mathcal{D}$ with the SGD optimizer. Let H denote the Hessian and G indicate the gradient, respectively. Specifically, we have $H^t_{\mathcal{D}} = \nabla^2_\theta \mathcal{L}(\mathcal{D}, \theta^t_{\mathcal{D}})$ and $H^t_{\mathcal{Z}} = \nabla^2_\theta \mathcal{L}(\mathcal{Z}, \theta^t_{\mathcal{D}})$, moreover, $G^t_{\mathcal{D}} = \nabla_\theta \mathcal{L}(\mathcal{D}, \theta^t_{\mathcal{D}})$ and $G^t_{\mathcal{Z}} = \nabla_\theta \mathcal{L}(\mathcal{Z}, \theta^t_{\mathcal{D}})$. For $\mathcal{Z} \subset \mathcal{D}$, the removal influence and the addition influence could be efficiently estimated via:*

$$\mathcal{I}_{-\mathcal{Z}} \approx -\sum_t \frac{\eta_t \sum_{k \geq t} \eta_k}{|\mathcal{D}|} \Big(H^t_{\mathcal{D}} G^t_{\mathcal{Z}} + H^t_{\mathcal{Z}} G^t_{\mathcal{D}} \Big), \quad (2)$$

$$\mathcal{I}_{+\mathcal{Z}} \approx \sum_t \frac{\eta_t \sum_{k \geq t} \eta_k}{|\mathcal{D}|} \Big(H^t_{\mathcal{D}} G^t_{\mathcal{Z}} + H^t_{\mathcal{Z}} G^t_{\mathcal{D}} \Big). \quad (3)$$

The estimator leverages parameter checkpoints along the SGD training trajectory t on $\mathcal{D}$. Although second-order in form, the Hessian-gradient product can be efficiently approximated via the established technique [39]:

$$HG \approx \lim_{\epsilon \to 0} \frac{\Big(\nabla_\theta \mathcal{L}(\theta + \epsilon G) - \nabla_\theta \mathcal{L}(\theta) \Big)}{\epsilon}, \quad (4)$$

where the computational complexity is $\mathcal{O}(p)$, with p denoting the number of parameters. This can be executed efficiently using popular deep learning frameworks [1]. We further establish a theoretical upper bound on the approximation error between our estimator and the exact influence obtained through full retraining. This bound guarantees robustness even under worst-case conditions and demonstrates strong practical reliability compared with prior estimators [26, 29, 41].

Theorem 2 ***(Error bound)*** *Let T denote the maximum iteration. By supposing the gradient of the loss is ℓ-Lipschitz continuous and the gradient norm of the network parameter is upper-bounded by g, and denote the maximum learning rate by η_{max}. The approximation error between the estimator ($\tilde{\mathcal{I}}$) and the exact one (denoted by $\mathcal{I}$) is bounded by:*

$$|\tilde{\mathcal{I}} - \mathcal{I}| \leq 2T^3 \ell (T+1) \eta_{max} g + \frac{|\mathcal{Z}|}{|\mathcal{D}|} T^2 g. \quad (5)$$

From this theorem, several observations follow: (i) The Lipschitz constant ℓ and gradient norm g jointly control the estimation error– models with more stable (less rapidly changing) gradients yield tighter bounds. (ii) The bound scales polynomially with training steps T, improving upon earlier estimators [26, 44] whose error grows exponentially. (iii) Although the bound represents a worst-case scenario, empirical results indicate that the estimated influence correlates closely with the exact influence across realistic settings. This reliability stems from the incremental nature of SGD updates, which stabilizes training dynamics and maintains the estimator's fidelity even for long optimization horizons.

4.2. Dataset Distillation via Influence Matching

Based on the influence estimator above, we formulate dataset distillation as learning a synthetic dataset $\mathcal{S}$ that can substitute the full real dataset $\mathcal{D}$ in terms of its overall influence on the model parameters. Intuitively, the influence of adding the synthetic data $\mathcal{S}$ should ideally offset the influence of removing the entire real dataset $\mathcal{D}$. Formally, this can be expressed as minimizing the total parameter shift:

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} \Big\| \mathcal{I}_{-\mathcal{D}} + \mathcal{I}_{+\mathcal{S}} \Big\|, \quad (6)$$

where $|\cdot|$ denotes a vector norm (typically L_2), measuring the magnitude of the residual parameter difference. By minimizing this quantity, we ensure that the model trained on $\mathcal{S}$

Algorithm 1 Dataset distillation by influence matching (**Inf-Match**)

Input: $\mathcal{D}$: original dataset; $\mathcal{T} = \{(\theta^t_{\mathcal{D}}, \eta_t)|^T_{t=1}\}$: a training trajectory of a network trained on $\mathcal{D}$; and t_m: the number of sampled time-steps.
Initialization: Initialize the synthetic set $\mathcal{S} = \{(x_i, \hat{y}_i)|x_i \in \mathcal{D}\}$, where $\hat{y}_i = f(x_i; \theta^T_{\mathcal{D}})$ is the soft-label from the learned model θ^T.
for t from 0 to max_iteration **do**
    Randomly sample a minibatch $B_{\mathcal{S}} \subset \mathcal{S}$, $B_{\mathcal{D}} \subset \mathcal{D}$.
    Sample checkpoints $\{\theta^{t_1}_{\mathcal{D}}, ..., \theta^{t_m}_{\mathcal{D}}\}$ from m time steps from training trajectory $\mathcal{T}$.
    Compute the loss via Eq.(7) on sampled checkpoints and $B_{\mathcal{S}}$ and $B_{\mathcal{D}}$.
    Update $\mathcal{S}$ (for both images and soft labels) by minimizing Eq.(7) with gradient descent.
end for
Output: The learned synthetic set $\mathcal{S}$.

yields parameters closely aligned with those trained on $\mathcal{D}$–achieving outcome alignment without retraining.

Remark 1 *The objective function defined in Eq.(6) establishes a novel, principled optimization goal: it aims to learn a synthetic dataset $\mathcal{S}$ such that the resulting model trained on $\mathcal{S}$ achieves an outcome (parameter update) that aligns with the outcome achieved by training on the real dataset $\mathcal{D}$. According to the additivity of influence functions [4, 28, 62], as shown by the identity $\left\|\mathcal{I}_{-\mathcal{D}} + \mathcal{I}_{+\mathcal{S}}\right\| = \left\|\left(\theta^*_{\mathcal{D}} + \mathcal{I}_{-\mathcal{D}} + \mathcal{I}_{+\mathcal{S}}\right) - \theta^*_{\mathcal{D}}\right\|$, minimizing this term is equivalent to minimizing the displacement between the model parameters $\theta^*_{\mathcal{D}}$ (trained on $\mathcal{D}$) and the parameters obtained after removing $\mathcal{D}$ and adding $\mathcal{S}$. This directly links the optimization to matching the final model outcomes, ensuring a robust, outcome-oriented data synthesis process.*

Objective Formulation of Inf-Match By substituting the estimator in Eq. (3) and Eq. (2) into the objective formulation defined in Eq. (6), we obtain a differentiable objective for dataset distillation,

$$\begin{aligned} \mathcal{S}^* &= \arg\min_{\mathcal{S}} \mathcal{J}(\mathcal{S}) \\ \text{s.t. } \mathcal{J}(\mathcal{S}) &= \Big\| - \sum_t \frac{2\eta_t \sum_{k\geq t} \eta_k}{|\mathcal{D}|} \left(H^t_{\mathcal{D}} G^t_{\mathcal{D}}\right) \\ &\quad + \sum_t \frac{\eta_t \sum_{k\geq t} \eta_k}{|\mathcal{D}|} \left(H^t_{\mathcal{D}} G^t_{\mathcal{S}} + H^t_{\mathcal{S}} G^t_{\mathcal{D}}\right)\Big\|, \end{aligned} \tag{7}$$

where $H^t_{\mathcal{D}}$ and $H^t_{\mathcal{S}}$ are the hessian of the parameter at the t-th (real-set) training iteration on the real and the synthetic set respectively, specifically, $H^t_{\mathcal{D}} = \nabla^2_\theta \mathcal{L}(\mathcal{D}, \theta^t_{\mathcal{D}})$ and $H^t_{\mathcal{S}} = \nabla^2_\theta \mathcal{L}(\mathcal{S}, \theta^t_{\mathcal{D}})$. And $G^t_{\mathcal{D}}$ and $G^t_{\mathcal{S}}$ are the gradient of the parameter at the t-th (real-set) training iteration on the real and the synthetic set respectively, specifically, $G^t_{\mathcal{D}} = \nabla_\theta \mathcal{L}(\mathcal{D}, \theta^t_{\mathcal{D}})$ and $G^t_{\mathcal{S}} = \nabla_\theta \mathcal{L}(\mathcal{S}, \theta^t_{\mathcal{D}})$. As for the efficient calculation for the Hessian-gradient production, please refer to Eq. (4).

During distillation, we do not compute gradients or Hessians over the entire dataset. Instead, we randomly sample mini-batches $B_{\mathcal{D}} \subset \mathcal{D}$ and $B_{\mathcal{S}} \subset \mathcal{S}$ to estimate these quantities, which greatly improves computational efficiency while preserving unbiased gradient estimates.

Inf-Match Algorithm The overall pipeline of **Inf-Match** is outlined in Alg. 1. We first train a base model on the real dataset $\mathcal{D}$ for T iterations and record the checkpoints $(\theta^t_{\mathcal{D}}, \eta_t)t = 1^T$. The synthetic set $\mathcal{S}$ is initialized with real images from $\mathcal{D}$ following the IPC (images-per-class) setting. At each iteration, we update both the synthetic images and labels by minimizing $\mathcal{J}(\mathcal{S})$ in Eq. (7) via gradient descent. To reduce memory consumption, we compute the loss using random mini-batches $B_{\mathcal{S}} \subset \mathcal{S}$ and $B_{\mathcal{D}} \subset \mathcal{D}$ from the synthetic and real datasets, respectively. After convergence, we output the learned synthetic set $\mathcal{S}$. Below, we detail the initialization and time-step sampling strategies used during training.

First, to initialize $\mathcal{S}$, we sample real images from $\mathcal{D}$ according to the IPC (Image Per Class) configuration and assign each a soft label produced by the final model $\theta^T_{\mathcal{D}}$. Both the synthetic images and labels are treated as learnable variables during training, following common practices in dataset distillation [5, 24, 55]. Compared with one-hot labels, soft labels allow inter-class information sharing, improving the representation efficiency of the distilled data.

Second, the loss in Eq. (7) involves averaging over all T training steps. To improve efficiency, we approximate this average by sampling m checkpoints at each update. We adopt a time-step selection schedule similar to DATM [24]: early in training, we sample earlier checkpoints to capture basic patterns, while in later stages, we shift toward later checkpoints to encode more complex, fine-grained structures. This progressive sampling strategy encourages structured learning from simple to difficult information, improving both convergence and final performance.

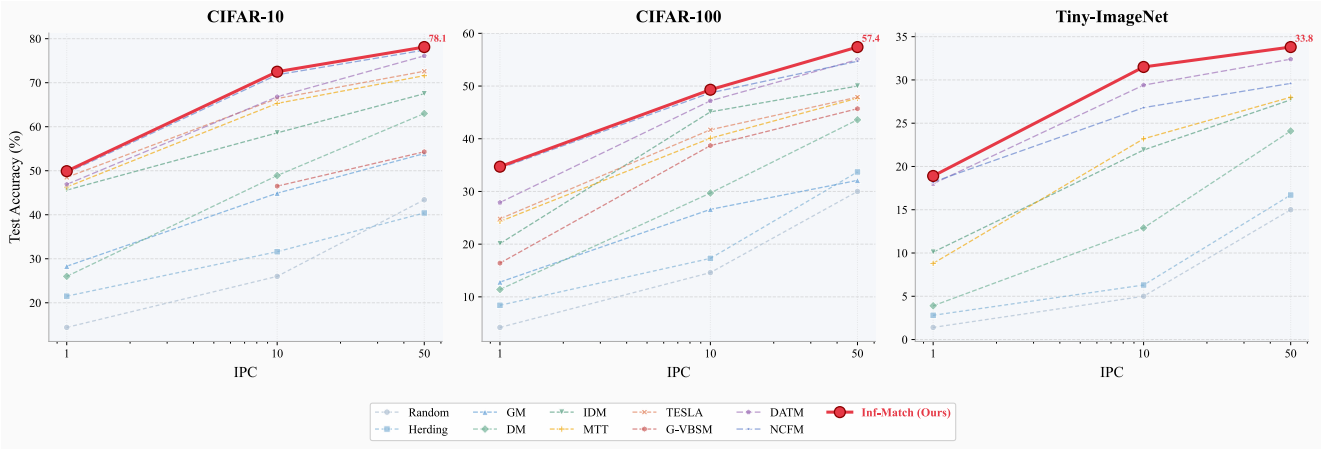


Figure 2. Performance comparison of different methods on CIFAR-10, CIFAR-100, and Tiny-ImageNet.

5. Experiments

This section presents the experimental evaluation of our Inf-Match on two scenarios: image classification and vision-language datasets. We first describe the experimental setups, including baselines and implementation details. We then report and analyze the experimental results on several datasets. Finally, we conduct an ablation study to investigate the effect of key components of our method.

5.1. Experimental Settings

Baselines. Here, we choose several baselines as competitors to our approach, including DD [55], GM [68], MTT [7], DM [66], IDM [69], DATM [24], TESLA [13], G-VBSM [46], NCFM [54], and also the famous coreset selection method Herding [37], and the popular baseline random selection.

Network Architecture. Our experiments default to the ConvNet architecture, consisting of three (four for Tiny ImageNet) conv-blocks (128 filters, pooling, ReLU, normalization) and a linear classifier. We also investigated LeNet [32], AlexNet [30], VGG11 [35], and ResNet18 [27] for cross-architecture comparisons. For vision-language tasks, we utilized a pre-trained, trainable Vision Transformer [18] and a frozen BERT [16] as backbones; both were independently pre-trained on unimodal data and connected via a trainable linear projection layer.

Setups. We evaluate our method under three different settings of Images Per Class (IPC): 1, 10, and 50. For vision-language datasets, we assess our approach across three synthetic dataset sizes: 100, 500, and 1000 samples. Each experiment consists of two distinct phases. First, we learn from the synthetic data and subsequently train a model on it to evaluate its performance on the real test set. All experiments are conducted using the PyTorch framework on a computing server with 8*A100 GPUs. Throughout all experiments, we set the batch size to 50. The optimizer employed in our experiments is SGD-M, with a momentum parameter configured at 0.9. For synthetic images, we set the corresponding learning rate as a value of 50.0, while we set the learning rate for the soft labels as 7.0. To ensure the robustness of our findings and the fairness of the evaluation, each experiment is independently repeated 10 times.

5.2. Image Classification

We conduct experiments for image classification datasets like CIFAR-10, CIFAR-100, and Tiny-ImageNet. The CIFAR-10 dataset [2] consists of 60,000 images across 10 different classes, while the CIFAR-100 dataset [2] is an extension of CIFAR-10, also containing 60,000 32×32-pixel color images divided into 100 fine-grained classes. The Tiny-ImageNet [60] dataset, containing 100000 training images, is a subset of the ImageNet dataset. It provides a more challenging dataset than the CIFAR series regarding the number of classes and the complexity of the images.

5.2.1. Main results

Figure 2 presents the impressive performance of our proposed method ("Ours") compared to various dataset distillation techniques across the CIFAR-10, CIFAR-100, and Tiny-ImageNet benchmarks. Our Inf-Match approach consistently achieves the leading results across every configuration of Images Per Class (IPC), establishing the effectiveness and robustness of our outcome-alignment strategy. For **CIFAR-10**, our method achieves the highest accuracy in all settings. It reaches $\mathbf{72.5}\%$ at IPC=10 and $\mathbf{78.1}\%$ at IPC=50, securing an advantage of approximately 0.7% over the previously top-performing NCFM [54]. Our $\mathbf{49.9}\%$ accuracy at IPC=1 also represents a new high watermark for extremely low IPC settings. Similarly, on **CIFAR-100**, our method maintains its dominance, leading all competitors with $\mathbf{49.3}\%$ at IPC=10 and a significant $\mathbf{57.4}\%$ at IPC=50. This superior perfor-

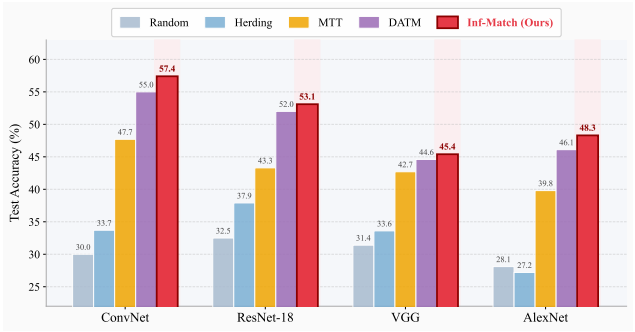


Figure 3. The cross-architecture generalization test on CIFAR-100 with the setting of IPC=50.

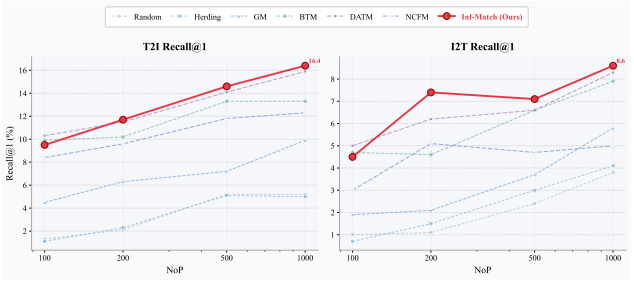


Figure 4. Performance comparison of different dataset distillation methods on the vision-language dataset, Flickr30K [40]. NoP means the number of (image-text) pairs. T2I and I2T indicate text-to-image and image-to-text, respectively.

mance is particularly notable at IPC=50, where we surpass NCFM's result by a margin of $\mathbf{2.7}\%$. The most pronounced gains are observed on **Tiny-ImageNet**, where our method delivers consistently superior performance across the board. At IPC=10 and IPC=50, our method achieves $\mathbf{31.5}\%$ and $\mathbf{33.8}\%$ respectively, outpacing NCFM by substantial margins of approximately **4.7%** and **4.2%**. Collectively, these results underscore the power of directly matching model influence, demonstrating our method's ability to deliver the best performance across various datasets and compression rates.

5.2.2. Generalization Evaluation

We subsequently assessed the cross-network generalization capability of the synthetic data generated by our method on the CIFAR-100 dataset using an IPC setting of 50. See Table 3 for details. Importantly, we observe that all distillation schemes presented in the table significantly surpass the performance of selection-based methods, such as random selection and Herding selection [37], in terms of cross-structure generalization. When compared with distillation-based methods (MTT [7] and DATM [24]). Across various model architecture configurations, our approach consistently outperformed the previous best method, DATM [24], achieving scores between 45.4% and 57.4%. This notable improvement underscores the efficacy of our method in producing synthetic data that generalizes effectively across diverse network architectures.

5.3. Vision-language Datasets

The field of multimodality [10, 34, 56, 57, 59, 70] has also made significant progress recently. We also conduct experiments for the well-known vision-language dataset Flickr30K [40], which comprises 31,783 images depicting daily activities and scenes sourced from the website. Each image is paired with five textual descriptions. We follow the experimental settings in BTM [58], where we choose the Normalizer-free ResNet [6] as the vision encoder and the BERT [16] model as the text encoder. While both encoders are pretrained, they are trained only on unimodal data and have no exposure to the other modality. A trainable linear projection layer with random initialization follows each encoder. The training loss for the model and the vision language dataset encourages the similarities between those paired images and texts [58]. We selected BTM [58] as the baseline. BTM introduces a bi-trajectory matching loss for dataset distillation in vision-language datasets, building upon the well-known trajectory matching technique (MTT [7]). As a result, we can readily adapt the updated version of the MTT algorithm, for example, DATM (difficulty-aware trajectory matching) [24], to BTM. Additionally, we have chosen several other baselines, including random selection, Herding selection [37], and GM [68].

We provide the experimental results in Figure 4. In most settings, our method consistently outperformed the others across all configurations. For instance, with 200 samples, our method achieved an impressive score of 7.4% on image-to-text retrieval tasks, higher than the next best, DATM [24], at 1.3%. With 500 samples, our method recorded 14.6% on the text-to-image setting, leading the pack, while the second-best competitor, DATM, managed only 14.1%. Finally, at 1000 samples, we maintained our superiority with a score of 16.4% on the text-to-image setting, surpassing all other methods. Another noteworthy observation from the experiments is that all dataset distillation methods outperformed the coreset methods based on selection. This finding aligns with previous results in Figure 2 and strongly indicates that dataset distillation is a promising direction for further research. Overall, our approach demonstrates robust performance across various sample sizes, highlighting its effectiveness on vision-language datasets.

5.4. Ablation study

Table 1 details the ablation study of our Inf-Match framework on CIFAR-100 (IPC $= 50$), confirming the contribution of each component while validating the strength of our core methodology. The sequential incorporation of enhancements

Table 1. Ablation study on CIFAR-100 with IPC=50. We also select NCFM [54], DATM[24], DM [66], GM [68], MTT [7] as baselines.

Real-data Init.	Learnable-Label	Sampling-Schedule	Performance
×	×	×	52.2
	GM [68]		32.1
	DM [66]		43.6
	MTT [7]		47.7
✓	×	×	53.7
✓	×	✓	55.0
✓	✓	×	54.6
✓	✓	✓	**57.4**
	DATM [24]		55.0
	NCFM [54]		54.7

improves performance: introducing Real-data Initialization boosts accuracy to 53.7%, utilizing the Sampling Schedule achieves 55.0%, and including Learnable Labels reaches 54.6%. When all components are integrated, our method attains its final accuracy of $\mathbf{57.4}\%$. This not only confirms the positive synergistic effect of our components but also robustly outperforms other advanced distillation methods like DATM ($\mathbf{55.0}\%$) and NCFM ($\mathbf{54.7}\%$), underscoring that our outcome-alignment strategy yields a superior solution space compared to process-matching techniques.

5.5. Learning process visualization

In Figure 5, we also conducted a comparative visualization of the learning process from synthetic data during algorithm execution. Firstly, we compare the performance as iterations progress. It is evident that, compared to the heuristic-based proxy task, *e.g.* MTT [7], our approach, which directly optimizes the original problem, exhibits a slower convergence rate but ultimately achieves significantly better performance. Secondly, we found that the fidelity of the synthetic images is not necessarily correlated with the performance of the final synthetic data. This observation arises from our findings that during the learning process, the images synthesized by our approach undergo a transition from increased realism to a subsequent increase in noise.

5.6. Feature space visualization

Furthermore, we compare the learned condensed dataset in the embedding space to study their distributions, see Fig. 6. We visualized the feature space of the "*Wolf*" category in CIFAR-100 under the IPC=10 setting. The white scatter points represent the projections of synthetic samples within this feature space. We found that when using DM [66] with the goal of feature distribution matching at small IPC settings (IPC=10), the synthetic samples produced tend to be overly concentrated in the high-density regions of the real data. In contrast, our approach generates synthetic samples that effectively occupy both high-density areas and the less dense

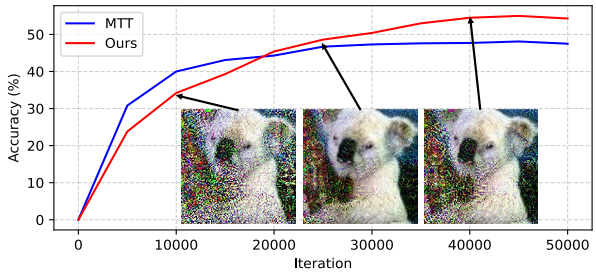
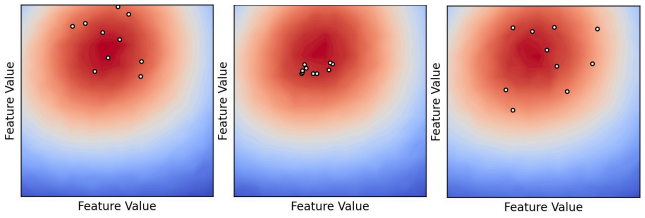


Figure 5. Visualization of the learning curve (MTT [7] v.s. Ours) and the intermediate synthetic results of ours. This experiment is conducted on CIFAR-100 with the setting of IPC=50.



(a) MTT (b) DM (c) Ours

Figure 6. An illustration of why our method performs well for dataset distillation is provided by comparing the feature distribution of our method and the classic DM (distribution matching) [66] and MTT (matching training trajectory) [7].

regions at the distribution's edges, demonstrating a more balanced representation compared to MTT and DM.

6. Conclusion

We introduced a novel dataset distillation framework that redefines the optimization objective from heuristic process alignment to rigorous outcome alignment. By developing an efficient, accurate, and fully differentiable sample influence estimator, we were able to precisely quantify the contribution of both the real and synthetic datasets to the final model parameters. Extensive experiments demonstrated the superior performance over state-of-the-art distillation methods.

7. Acknowledgment

The work has been supported by Hong Kong Research Grant Council-General Research Fund Scheme (Grant No. 17202422, 17212923, 17215025), Theme-based Research (Grant No. T45-701/22-R), and Strategic Topics Grant (Grant No. STG3/E-605/25-N). Part of the described research work is conducted in the JC STEM Lab of Robotics for Soft Materials funded by The Hong Kong Jockey Club Charities Trust.

References

[1] Adam Paszke, Sam Gross, Soumith Chintala, G Chanan, E Yang, Zachary Devito, Zeming Lin, Alban Desmaison, L Antiga, A Lerer, and et.al. Automatic differentiation in pytorch. In *Advances in neural information processing systems Workshop*, 2017. 4

[2] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical report*, 2009. 6

[3] Alexander Atanasov, Blake Bordelon, and Cengiz Pehlevan. Neural networks as kernel learners: The silent alignment effect. *arXiv preprint arXiv:2111.00034*, 2021. 3

[4] Samyadeep Basu, Xuchen You, and Soheil Feizi. On second-order group influence functions for black-box predictions. In *International Conference on Machine Learning*, 2020. 1, 3, 5

[5] Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales. Flexible dataset distillation: Learn labels instead of images. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Workshop*, 2020. 5

[6] Andy Brock, Soham De, Samuel L. Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. In *Proceedings of the 38th International Conference on Machine Learning*, pages 1059–1071. PMLR, 2021. 7

[7] 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, 6, 7, 8

[8] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Generalizing dataset distillation via deep generative prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3739–3748, 2023. 3

[9] Siu Lun Chau, Robert Hu, Javier Gonzalez, and Dino Sejdinovic. Rkhs-shap: Shapley values for kernel methods. *Advances in neural information processing systems*, 35:13050–13063, 2022. 3

[10] Yingxian Chen, Jiahui Liu, Ruidi Fan, Yanwei Li, Chirui Chang, Shizhen Zhao, Wilton WT Fok, Xiaojuan Qi, and Yik-Chung Wu. Aligning effective tokens with video anomaly in large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22695–22706, 2025. 7

[11] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gerard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. *Journal of Machine Learning Research*, 38: 192–204, 2015. 3

[12] R Dennis Cook. Assessment of local influence. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 48(2):133–155, 1986. 3

[13] 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. 2, 3, 6

[14] Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems*, pages 2933–2941, 2014. 3

[15] Zhiwei Deng and Olga Russakovsky. Remember the past: Distilling datasets into addressable memories for neural networks. *Advances in Neural Information Processing Systems*, 35:34391–34404, 2022. 3

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 6, 7

[17] 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

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 6

[19] 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. 2, 3

[20] Jiawei Du, Qin Shi, and Joey Tianyi Zhou. Sequential subset matching for dataset distillation. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[21] Yunzhen Feng, Shanmukha Ramakrishna Vedantam, and Julia Kempe. Embarrassingly simple dataset distillation. In *The Twelfth International Conference on Learning Representations*, 2023. 2

[22] Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, et al. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*, 2023. 1, 3, 4

[23] Jianyang Gu, Kai Wang, Wei Jiang, and Yang You. Summarizing stream data for memory-restricted online continual learning. *arXiv preprint arXiv:2305.16645*, 2023. 1

[24] 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. 1, 2, 3, 5, 6, 7, 8

[25] Zayd Hammoudeh and Daniel Lowd. Training data influence analysis and estimation: A survey. *arXiv preprint arXiv:2212.04612*, 2022. 3

[26] Satoshi Hara, Atsushi Nitanda, and Takanori Maehara. Data cleansing for models trained with sgd. *Advances in Neural Information Processing Systems*, 32, 2019. 4

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE / CVF Computer Vision and Pattern Recognition Conference*, 2016. 6

[28] Pang Koh, Kai-Siang Ang, Hubert H. K. Teo, and Percy Liang. On the accuracy of influence functions for measuring group effects. In *Advances in neural information processing systems*, 2019. 1, 3, 4, 5

[29] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, 2017. 1, 3, 4

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the International Conference on Neural Information Processing Systems*, page 1097–1105, 2012. 6

[31] Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models. *arXiv preprint arXiv:2310.00902*, 2023. 1, 3, 4

[32] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 6

[33] Guang Li, Bo Zhao, and Tongzhou Wang. Awesome dataset distillation. `https://github.com/Guang000/Awesome-Dataset-Distillation`, 2022. 1

[34] Jiahui Liu, Chirui Chang, Jianhui Liu, Xiaoyang Wu, Lan Ma, and Xiaojuan Qi. Mars3d: A plug-and-play motion-aware model for semantic segmentation on multi-scan 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9372–9381, 2023. 7

[35] Shuying Liu and Weihong Deng. Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 730–734, 2015. 6

[36] Noel Loo, Ramin Hasani, Alexander Amini, and Daniela Rus. Efficient dataset distillation using random feature approximation. *Advances in Neural Information Processing Systems*, 35:13877–13891, 2022. 3

[37] Max Welling. Herding dynamical weights to learn. In *International Conference on Machine Learning*, 2009. 6, 7

[38] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. In *International Conference on Learning Representations*, 2020. 3

[39] Barak A. Pearlmutter. Fast exact multiplication by the hessian. *Neural Computation*, 6(1):147–160, 1994. 4

[40] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015. 7

[41] Garima Pruthi, Frederick Liu, Sundararajan Mukund, and Satyen Kale. Estimating training data influence by tracing gradient descent. *arXiv preprint arXiv:2002.08484*, 2020. 4

[42] 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

[43] Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. Scaling up influence functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. 1, 3, 4

[44] Andrea Schioppa, Katja Filippova, Ivan Titov, and Polina Zablotskaia. Theoretical and practical perspectives on what influence functions do. *Advances in Neural Information Processing Systems*, 36, 2024. 4

[45] Lloyd S Shapley et al. A value for n-person games. 1953. 3

[46] Xindong Zhang Shitong Shao, Zeyuan Yin and Zhiqiang Shen. Generalized large-scale data condensation via various backbone and statistical matching. *arXiv preprint arXiv:2311.17950*, 2023. 6

[47] Ilia Sucholutsky and Matthias Schonlau. Soft-label dataset distillation and text dataset distillation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021. 3

[48] Peng Sun, Bei Shi, Daiwei Yu, and Tao Lin. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. *arXiv preprint arXiv:2312.03526*, 2023. 3

[49] Haoru Tan, Sitong Wu, Fei Du, Yukang Chen, Zhibin Wang, Fan Wang, and Xiaojuan Qi. Data pruning via moving-one-sample-out. In *Advances in neural information processing systems*, 2023. 3

[50] Haoru Tan, Sitong Wu, Xiuzhe Wu, Wang Wang, Bo Zhao, Zeke Xie, Gui-Song Xia, and Xiaojuan Qi. Understanding data influence with differential approximation, 2025.

[51] Haoru Tan, Xiuzhe Wu, Sitong Wu, Shaofeng Zhang, Yanfeng Chen, Xingwu Sun, Jeanne Shen, and XIAOJUAN QI. Understanding data influence in reinforcement finetuning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 3

[52] 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

[53] Kai Wang, Jianyang Gu, Daquan Zhou, Zheng Zhu, Wei Jiang, and Yang You. Dim: Distilling dataset into generative model. *arXiv preprint arXiv:2303.04707*, 2023. 3

[54] Shaobo Wang, Yicun Yang, Zhiyuan Liu, Chenghao Sun, Xuming Hu, Conghui He, and Linfeng Zhang. Dataset distillation with neural characteristic function: A minmax perspective, 2025. 1, 2, 6, 8

[55] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. 1, 3, 5, 6

[56] Sitong Wu, Haoru Tan, Zhuotao Tian, Yukang Chen, Xiaojuan Qi, and Jiaya Jia. Saco loss: Sample-wise affinity consistency for vision-language pre-training. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27348–27359, 2024. 7

[57] Sitong Wu, Haoru Tan, Yukang Chen, Shaofeng Zhang, Jingyao Li, Bei Yu, Xiaojuan Qi, and Jiaya Jia. Mixture-of-scores: Robust image-text data valuation via three lines of code. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 24603–24614, 2025. 7

[58] Xindi Wu, Byron Zhang, Zhiwei Deng, and Olga Russakovsky. Multimodal dataset distillation for image-text retrieval. *arXiv preprint arXiv:2308.07545*, 2023. 7

[59] Bin Xia, Bohao Peng, Yuechen Zhang, Junjia Huang, Jiyang Liu, Jingyao Li, Haoru Tan, Sitong Wu, Chengyao Wang, Yitong Wang, Xinglong Wu, Bei Yu, and Jiaya Jia. Dreamomni2: Multimodal instruction-based editing and generation, 2025. 7

[60] Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 2015. 6

[61] 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

[62] Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence. In *International Conference on Learning Representations*, 2023. 5

[63] Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[64] David Junhao Zhang, Heng Wang, Chuhui Xue, Rui Yan, Wenqing Zhang, Song Bai, and Mike Zheng Shou. Dataset condensation via generative model. *arXiv preprint arXiv:2309.07698*, 2023. 3

[65] 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

[66] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. *arXiv preprint arXiv:2110.04181*, 2021. 1, 2, 3, 6, 8

[67] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021. 1, 2, 3

[68] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2020. 1, 2, 3, 6, 7, 8

[69] Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation. 2023. 2, 6

[70] Shizhen Zhao, Jiahui Liu, Xin Wen, Haoru Tan, and Xiaojuan Qi. Equipping vision foundation model with mixture of experts for out-of-distribution detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1751–1761, 2025. 7

[71] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. *Advances in Neural Information Processing Systems*, 35:9813–9827, 2022. 3