Optimizing Distributional Geometry Alignment with Optimal Transport for Generative Dataset Distillation

Xiao Cui[1,2] **Yulei Qin**[3] **Wengang Zhou**[1] **Hongsheng Li**[2] **Houqiang Li**[1]
[1] University of Science and Technology of China
[2] CUHK MMLab [3] Independent Researcher
cuixiao2001@mail.ustc.edu.cn, yuleichin@126.com
{zhwg,lihq}@ustc.edu.cn, hsli@ee.cuhk.edu.hk

Abstract

Dataset distillation seeks to synthesize a compact distilled dataset, enabling models trained on it to achieve performance comparable to models trained on the full dataset. Recent methods for large-scale datasets focus on matching global distributional statistics (e.g., mean and variance), but overlook critical instance-level characteristics and intraclass variations, leading to suboptimal generalization. We address this limitation by reformulating dataset distillation as an Optimal Transport (OT) distance minimization problem, enabling fine-grained alignment at both global and instance levels throughout the pipeline. OT offers a geometrically faithful framework for distribution matching. It effectively preserves local modes, intra-class patterns, and fine-grained variations that characterize the geometry of complex, high-dimensional distributions. Our method comprises three components tailored for preserving distributional geometry: (1) OT-guided diffusion sampling, which aligns latent distributions of real and distilled images; (2) label-image-aligned soft relabeling, which adapts label distributions based on the complexity of distilled image distributions; and (3) OT-based logit matching, which aligns the output of student models with soft-label distributions. Extensive experiments across diverse architectures and large-scale datasets demonstrate that our method consistently outperforms state-of-the-art approaches in an efficient manner, achieving at least 4% accuracy improvement under IPC=10 settings for each architecture on ImageNet-1K.

1 Introduction

The expansion of data has fueled advances in deep learning, but also introduced prohibitive costs in storage, computation, and energy [1, 2, 3]. To address these challenges, dataset distillation aims to synthesize a small set of training samples to expedite model training while maintaining comparable performance [4]. Such distillation not only improves accessibility and cost-efficiency, but also facilitates practical applications such as knowledge transfer [5], federated learning [6, 7], and continual learning [8, 9]. Moreover, it provides a valuable lens to investigate the theoretical principles underlying training efficiency and representation capacity in deep learning [10, 11].

Traditional dataset distillation methods can be broadly categorized into optimization-based [12, 13, 14, 15] and distribution-matching-based approaches [16, 17, 18, 19]. Despite their effectiveness, these methods remain largely restricted to small-scale, low-resolution datasets such as MNIST [20], CIFAR [21], or downsampled ImageNet subsets [22]. This limitation stems from the prohibitive computational cost of alternating optimization between the distilled data and the condensation

Corresponding authors: Wengang Zhou and Houqiang Li.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

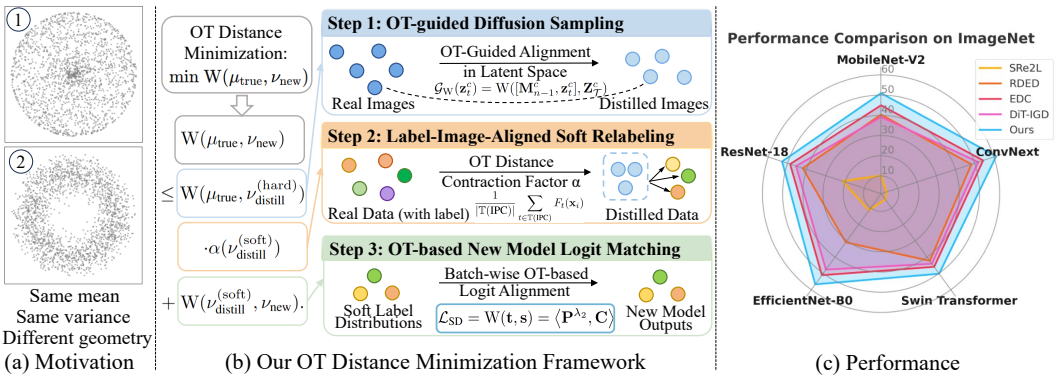


Figure 1: (a) Distributions with identical mean or variance may differ geometrically, causing biases in global-statistics-based optimization. (b) We decompose the OT objective into three stages: OT-guided diffusion sampling, label-image-aligned soft relabeling, and OT-based logit matching. (c) Our method consistently outperforms prior approaches across architectures on ImageNet-1K (IPC = 10).

model [23], and the reliance on integrated image representations that demand costly pixel-level refinement [24]. Recent efforts have explored generative and model-inversion-based techniques to overcome these scalability bottlenecks. Model-inversion-based methods [25, 26, 27, 28], originally proposed under data-free distillation framework [29, 30], rely entirely on global batch-normalization statistics extracted from pretrained models. While simple, this design imposes an inherent limitation: it fundamentally lacks the ability to recover or preserve instance-level, local distributional structures. In contrast, generative-model-based methods [24, 31, 32, 33] leverage real image samples during the sampling process, showing potential to approximate the true data distribution more faithfully.

However, existing generative approaches have yet to fully realize this promise, as they still solely focus on matching global gradient statistics. Besides, the fine-grained distributional structures are not properly captured by cosine-similarity-based diversity guidance, resulting in local mode collapse and distributional mismatch in the distilled set. To address this limitation, we propose a principled reformulation grounded in Optimal Transport (OT), which enables fine-grained distributional geometry alignment between real distribution and model output distribution. Specifically, we define *distributional geometry alignment* as preserving distribution-level global and local structures (e.g., coarse-grained patterns and subclass densities), rather than image-level features. Our key insight is that each real data point encapsulates rich intra-class semantic variation, such as the distinctive traits of different subclasses or local modes within the same class. OT inherently provides a geometrically faithful and perceptually aligned measure of distributional differences [34], making it especially promising for preserving and transferring these fine-grained semantic structures during distillation [35].

Building upon this, we formulate dataset distillation as an OT distance minimization problem. As shown in Figure 1, to make the alignment process tractable and optimization-friendly, we decompose the total OT distance into three complementary objectives that altogether contribute to its minimization: (1) instance-level transport in image latent space, (2) label-image alignment in label space, and (3) batch-wise logit alignment between new model predictions and soft targets. Such decomposition ensures alignment through the sequential stages of dataset distillation, ranging from image generation and label assignment to student model training. In the first stage, latent space transport is achieved by continuously computing the OT distance between the accumulated synthetic images (including the newly generated samples) and the real image batches at each sampling step. The gradients from this computation are used to guide the diffusion sampling process. In the second stage, we align the complexity of the synthetic image distribution with that of the soft label distribution, thereby narrowing the OT distance between the distilled data and the real data. In the final stage, we transfer the rich distributional geometric information embedded in the distilled set to new student models by minimizing the batch-wise OT distance between the student outputs and the soft-label distributions.

We evaluate our method across a diverse range of architectures, including ResNet, MobileNet, EfficientNet, Swin Transformer, ConvNet, and ConvNeXt. Our approach consistently outperforms

state-of-the-art methods across all datasets, architectures, and IPCs, achieving particularly strong results on ImageNet-1K [22] under the challenging IPC=10 scenario. Our contributions are threefold:

- We propose a novel perspective of dataset distillation by formulating the task as an OT distance minimization problem. We decompose the objective into three tractable components.
- We systematically enhance distributional geometry alignment through key stages of the pipeline, including image sampling, soft label relabeling, and student model logit matching.
- We demonstrate the effectiveness and generalizability of our method across a broad range of datasets and model architectures, which significantly surpass existing techniques.

2 Related Works

Numerous studies have investigated dataset distillation: initial works target low-resolution, small-scale datasets, while more recent methods address large-scale, higher-resolution scenarios.

2.1 Small-scale distillation methods

Traditional dataset distillation methods can be broadly classified into optimization-based and distribution-matching (DM)-based approaches. Optimization-based methods [12, 13, 36, 14, 15] adopt a bi-level optimization framework, where model parameters are updated in the outer loop while synthetic data are refined in the inner loop to match gradients or trajectories. In contrast, DM-based methods [16, 17, 18, 37, 38, 19] directly align the feature distributions of real and synthetic data, thereby avoiding costly nested optimization. However, all these methods exhibit high model dependence on the condensation model, which limits the versatility of the distilled datasets in generalizing across different architectures [39, 40]. Also, they incur significant time and memory costs due to three factors: (1) treating synthetic data as fixed entities, (2) requiring exhaustive pixel-level refinements, and (3) relying on real data for image refinement. As a result, traditional dataset distillation approaches are predominantly applied to small-scale datasets.

2.2 Large-scale distillation methods

Recent methods for large-scale, high-resolution datasets fall into two main categories: model-inversion-based and generative-model-based methods. Model-inversion-based methods [27, 28, 23, 41, 42, 43] compress the real dataset into a compact model representation, eliminating the need for real data during image refinement. This reduces memory overhead and allows scaling to large datasets such as ImageNet-1K [22]. However, the lack of real data in the reconstruction process results in the loss of fine-grained instance-level information, which hinders the distilled dataset from accurately capturing the structural properties and instance-specific characteristics of the real distribution. Generative-model-based methods [32, 44, 45, 46, 24] leverage pretrained generative models to avoid pixel-level refinements. While generating one sample at a time reduces memory overhead and avoids treating data as fixed entities, independently synthesizing each sample prevents the distilled dataset from maintaining a coherent overall distribution, thereby limiting its ability to capture the full diversity and structural relationships of the real data distribution. Due to their inherent exclusion of real data during reconstruction, model-inversion-based methods fail to preserve fine-grained structures of the real distribution; accordingly, we adopt the generative-model-based paradigm as our starting point. We address the shortcomings of both families by proposing an OT framework that ensures distributional geometry alignment throughout the distillation process.

3 Preliminaries

Given images $\mathbf{x} \sim q(\mathbf{x})$, define the induced latent distribution q_Z by $\mathbf{z}_0 = E(\mathbf{x})$, $\mathbf{z}_0 \sim q_Z(\mathbf{z_0})$, where E is the encoder mapping images into latent space. A latent diffusion model learns $p_\phi(\mathbf{z}_0) \approx q_Z(\mathbf{z}_0)$, from which we can efficiently sample. Let D be a decoder that reconstructs images via $\hat{\mathbf{x}} = D(\mathbf{z}_0)$. The forward noising process corrupts the clean latent $\mathbf{z}_0$ via Gaussian perturbations:

$$\mathbf{z}_t = \sqrt{\alpha_t}\mathbf{z}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \tag{1}$$

Table 1: Distributions involved in dataset distillation. All are defined over the joint space $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ denotes the image space and $\mathcal{Y}$ denotes the label space.

Distribution	Image Source	Label Source	Description
$\mu_{\text{true}}(\mathbf{x}, \mathbf{y})$	Real (full) Images	Ground-truth $y(\mathbf{x})$	True data-label distribution
$\nu_{\text{distill}}^{(\text{soft})}(\mathbf{x}, \mathbf{y})$	Distilled (generated) Images	Teacher soft label $\mathbf{t}(\mathbf{x})$	Distilled data with soft label
$\nu_{\text{distill}}^{(\text{hard})}(\mathbf{x}, \mathbf{y})$	Distilled (generated) Images	One-hot label $y_{\text{onehot}}(\mathbf{x})$	Distilled data with hard label
$\nu_{\text{new}}(\mathbf{x}, \mathbf{y})$	Distilled (generated) Images	Student logit output $\mathbf{s}(\mathbf{x})$	Output of the model trained on $\mathcal{S}$

where α_t controls the noise schedule. The reverse process reconstructs clean samples via a parameterized denoising function $\epsilon_\phi(\mathbf{z}_t, t)$, iteratively refining noisy inputs with update function s:

$$\mathbf{z}_{t-1} = s(\mathbf{z}_t, t, \epsilon_\phi(\mathbf{z}_t, t)). \tag{2}$$

Guided diffusion [47] modifies this process by introducing an auxiliary guidance function $\mathcal{G}(\mathbf{z}_t, t)$ that adjusts the sampling trajectory. This allows generation to be conditioned on labels, structural priors, or more abstract objectives by modifying $s(\mathbf{z}_t, t, \epsilon_\phi)$ to optimize for a task-specific constraint.

Influence-Guided Diffusion (IGD) [24] leverages guided diffusion for dataset distillation by modifying the reverse sampling process to generate training-optimal data. Instead of passively sampling from $p_\phi(\mathbf{x})$, IGD introduces trajectory influence function and diversity function into the diffusion process to prioritize samples. Given a guided diffusion framework, IGD modifies the sampling update as:

$$\mathbf{z}_{t-1} = s(\mathbf{z}_t, t, \epsilon_\phi) - \rho_t \nabla_{\mathbf{z}_t} \mathcal{G}_I(\mathbf{z_t}, t) - \gamma_t \nabla_{\mathbf{z}_t} \mathcal{G}_D(\mathbf{z}_t), \tag{3}$$

where $\mathcal{G}_I(\mathbf{z}_t, t)$ represents the influence function for global distributional trajectory matching, and $\mathcal{G}_D(\mathbf{z}_t)$ enforces diversity to prevent redundancy in the distilled dataset.

4 Methods

4.1 Problem Statement

Dataset distillation aims to construct a compact distilled dataset $\mathcal{S} \equiv \nu_{\text{distill}}(\mathbf{x}, \mathbf{y})$ (this means that ν_{distill} denotes the empirical distribution over dataset $\mathcal{S}$) from a real, full dataset $\mathcal{T} \equiv \mu_{\text{true}}(\mathbf{x}, \mathbf{y})$, such that $|\mathcal{S}| \ll |\mathcal{T}|$. A student model trained on $\mathcal{S}$ should mimic the performance of training on $\mathcal{T}$, i.e., the output distribution $\nu_{\text{new}}(\mathbf{x}, \mathbf{y})$ of the student model should remain close to the ground-truth distribution $\mu_{\text{true}}(\mathbf{x}, \mathbf{y})$. We formulate the dataset distillation objective as an OT minimization problem, where we minimize the Wasserstein distance $\mathrm{W}(\mu_{\text{true}}, \nu_{\text{new}})$ between the ground-truth and student-induced distributions. The key distributions involved in the formulations are summarized in Table 1. We provide a detailed description of symbols in Appendix D.

4.2 Reconstructing the Optimal Transport Distance

We now provide a theoretical decomposition of our objective, $\mathrm{W}(\mu_{\text{true}}, \nu_{\text{new}})$, by introducing two key principles: (1) the triangle inequality partitioning the discrepancy introduced before and after distilled set construction, and (2) a multiplicative contraction term reflecting the benefit of soft labels over hard labels. Unlike the commonly used measures such as KL divergence and cosine similarity, which do not satisfy the triangle inequality, the Wasserstein distance $\mathrm{W}(\cdot, \cdot)$ is a proper metric on the space of distributions. This property allows us to decompose the total discrepancy as:

$$\mathrm{W}(\mu_{\text{true}}, \nu_{\text{new}}) \leq \mathrm{W}(\mu_{\text{true}}, \nu_{\text{distill}}^{(\text{hard})}) + \mathrm{W}(\nu_{\text{distill}}^{(\text{hard})}, \nu_{\text{new}}). \tag{4}$$

As soft labels better approximate the true label distribution than one-hot assignments, recent works adopt soft labels instead of hard one-hot assignments, leading to the following relaxed upper bound:

$$\mathrm{W}(\mu_{\text{true}}, \nu_{\text{new}}) \leq \underbrace{\mathrm{W}(\mu_{\text{true}}, \nu_{\text{distill}}^{(\text{soft})})}_{\text{Dataset Discrepancy}} + \underbrace{\mathrm{W}(\nu_{\text{distill}}^{(\text{soft})}, \nu_{\text{new}})}_{\text{Logit Matching Error}}. \tag{5}$$

The first term captures the mismatch between the distilled and real data distribution. The second term reflects the logit-wise alignment between the student model's output distribution and the soft labels

of the distilled data. Directly minimizing the first term is challenging due to too many variables to optimize. To analyze it further, we model the soft label advantage via a multiplicative relation:

$$\mathrm{W}(\mu_{\text{true}}, \nu_{\text{distill}}^{(\text{soft})}) = \mathrm{W}(\mu_{\text{true}}, \nu_{\text{distill}}^{(\text{hard})}) \cdot \alpha(\nu_{\text{distill}}^{(\text{soft})}) \tag{6}$$

The contraction factor $\alpha(\nu_{\text{distill}}^{(\text{soft})})$ measures how much soft labels reduce the discrepancy between label and image distributions compared to hard labels alone. The contraction is achieved by matching the complexity of the teacher-provided soft label distributions to that of the distilled image distribution. Since both $\nu_{\text{distill}}^{(\text{hard})}$ and μ_{true} use one-hot (hard) labels, their Wasserstein distance can be computed class-wise, by independently solving optimal transport between images of the same category:

$$\mathrm{W}(\mu_{\text{true}}, \nu_{\text{distill}}^{(\text{hard})}) = \mathbb{E}_y \left[\mathrm{W}(\mu_{\text{true}}(\mathbf{x} \mid y), \nu_{\text{distill}}^{(\text{hard})}(\mathbf{x} \mid y) \right] \tag{7}$$

where $\mathbb{E}_y$ denotes the expectation over label classes, which measures the average conditional Wasserstein distance across classes. Putting everything together, we arrive at a structured upper bound:

$$\mathrm{W}(\mu_{\text{true}}, \nu_{\text{new}}) \leq \mathbb{E}_y \left[\mathrm{W}(\mu_{\text{true}}(\mathbf{x} \mid y), \nu_{\text{distill}}^{(\text{hard})}(\mathbf{x} \mid y) \right] \cdot \alpha(\nu_{\text{distill}}^{(\text{soft})}) + \mathrm{W}(\nu_{\text{distill}}^{(\text{soft})}, \nu_{\text{new}}) \tag{8}$$

where each term is controlled by a distinct design choice: OT-guided diffusion sampling, label-image-aligned soft label relabeling, and OT-based logit matching between the student model and the distilled dataset. This decomposition allows a principled basis of our method, which explicitly targets at minimization of each component. Our pseudocodes are provided in Appendix E.

4.3 OT-guided Diffusion Sampling (OTG)

In the remainder of this section, we optimize the three terms in Equation 8 sequentially. We now concentrate on the first term. For each class c, we minimize the class-conditional OT distance $\mathrm{W}(\mu_{\text{true}}(\mathbf{x} \mid c), \nu_{\text{distill}}^{(\text{hard})}(\mathbf{x} \mid c))$ through diffusion guidance: we compute the OT distance between the distilled images and the real images in the latent space as the guiding function. At each diffusion step during the generation of the n-th latent $\mathbf{z}_0^c$, we draw a random batch of class-c samples from dataset $\mathcal{T}$ and encode them into latent representations $\mathbf{Z}_{\mathcal{T}}^c$. We then employ the following guidance function:

$$\mathcal{G}_{\mathrm{W}}(\mathbf{z}_t^c) = \mathrm{W}([\mathbf{M}_{n-1}^c, \mathbf{z}_t^c], \mathbf{Z}_{\mathcal{T}}^c) \tag{9}$$

where $\mathbf{M}_{n-1}^c$ denotes previously sampled $n-1$ latents for class c, and $[\cdot]$ represents the pythonic concatenation.We denote $[\mathbf{M}_{n-1}^c, \mathbf{z}_t^c]$ as $\hat{\mathbf{M}}_n^c$. The OT matrix $\mathbf{P}^{\lambda_1}$ can be efficiently approximated:

$$\mathbf{P}^{\lambda_1} = \underset{\mathbf{P}}{\operatorname{argmin}} \left\langle \mathbf{P}, \mathbf{D}(\hat{\mathbf{M}}_n^c, \mathbf{Z}_{\mathcal{T}}^c) \right\rangle - \lambda_1 h(\mathbf{P}), \text{ where } \sum_i \mathbf{P}_{ij} = \frac{1}{|\mathbf{Z}_{\mathcal{T}}^c|} \; \forall j, \; \sum_j \mathbf{P}_{ij} = \frac{1}{n} \; \forall i. \tag{10}$$

where $h(\mathbf{P})$ is the entropy of $\mathbf{P}$, $\langle \cdot, \cdot \rangle$ denotes the Frobenius inner product, $\lambda_1 > 0$ is the entropy regularization weight, $\mathbf{D}(\hat{\mathbf{M}}_n^c, \mathbf{Z}_{\mathcal{T}}^c)$ represents the cost matrix measuring the pairwise distance between the real latent representations $\mathbf{Z}_{\mathcal{T}}^c$ and the sampled $\hat{\mathbf{M}}_n^c$. Without loss of generality, we use the ℓ_p-norm cost matrix and initialize the candidate transport matrix $\mathbf{K}^0$ as:

$$\mathbf{D}_{ij}(\hat{\mathbf{M}}_n^c, \mathbf{Z}_{\mathcal{T}}^c) = \parallel \hat{\mathbf{M}}_n^c[i] - \mathbf{Z}_{\mathcal{T}}^c[j] \parallel_p, \quad \mathbf{K}^0 = \exp(-\frac{\mathbf{D}}{\lambda_1}). \tag{11}$$

Next, Sinkhorn normalization is applied through iterative updates to $\mathbf{K}$:

$$\widehat{\mathbf{K}}^i \leftarrow \operatorname{diag}\left(\mathbf{K}^{i-1}\mathbf{1}_n \oslash (n\mathbf{1}_{|\mathbf{Z}_{\mathcal{T}}^c|})\right)^{-1} \mathbf{K}^{i-1}, \;\; \mathbf{K}^i \leftarrow \widehat{\mathbf{K}}^i \operatorname{diag}\left(\left(\widehat{\mathbf{K}}^i\right)^{\mathrm{T}} \mathbf{1}_{|\mathbf{Z}_{\mathcal{T}}^c|} \oslash (|\mathbf{Z}_{\mathcal{T}}^c|\mathbf{1}_n)\right)^{-1}, \tag{12}$$

where $\oslash$ denotes element-wise division, $(\cdot)^{\mathrm{T}}$ indicates matrix transpose. After T iterations, the optimal transport matrix $\mathbf{P}^{\lambda_1}$ is obtained, and we can compute the Sinkhorn distance (an approximation of the OT distance) W as:

$$\mathbf{P}^{\lambda_1} = \mathbf{K}^T, \quad \mathrm{W}([\mathbf{M}_{n-1}^c, \mathbf{z}_t^c], \mathbf{Z}_{\mathcal{T}}^c) = \left\langle \mathbf{P}, \mathbf{D}(\hat{\mathbf{M}}_n^c, \mathbf{Z}_{\mathcal{T}}^c) \right\rangle = \sum_{i,j} \mathbf{K}_{ij}^T \mathbf{D}_{ij} \tag{13}$$

For the entire guided diffusion sampling, we follow the previous approach to combine the terms of trajectory and diversity functions with our OT function. The iteration of $t = T_D \to 1$ yields $\mathbf{z}_0^c$:

$$\mathbf{z}_{t-1}^c = s(\mathbf{z}_t^c, t, \epsilon_\phi) - \rho_t \nabla_{\mathbf{z}_t^c} \mathcal{G}_I(\mathbf{z}_t^c, t) - \gamma_t \nabla_{\mathbf{z}_t^c} \mathcal{G}_D(\mathbf{z}_t^c) - \beta_1 \nabla_{\mathbf{z}_t^c} \mathcal{G}_{\mathrm{W}}(\mathbf{z}_t^c), \tag{14}$$

where ρ_t, γ_t and β_1 are weights, $\mathcal{G}_I(\mathbf{z}_t^c, t)$ and $\mathcal{G}_D(\mathbf{z}_t^c)$ are trajectory and diversity functions, respectively. By minimizing the OT distance in the image distribution space, we account for the contribution of individual real images and incorporate both global and local structural information, thereby promoting fine-grained geometric alignment between distributions. Finally, we use decoder D to convert all latent representations into images, forming the distilled image set $\mathcal{S}_{\mathbf{x}}$.

4.4 Label-Image-Aligned Soft Label Relabeling (LIA)

We now focus on the contraction factor $\alpha(\nu_{\text{distill}}^{(\text{soft})})$, which characterizes the alignment between the complexity of the soft label distribution and that of the distilled image distribution (Appendix G.2 for details). Since the representational capacity of the distilled dataset is primarily governed by the number of images per class (IPC), we adopt an IPC-aware strategy that minimizes the overall OT distance. In low-IPC regimes, the distilled image distribution is less expressive and more prone to overfitting. Assigning overly complex soft labels in such cases can introduce distributional mismatch and degrade alignment. To mitigate this, we employ a smaller number of representative teacher models to produce simplified, low-entropy soft labels that offer well-calibrated supervision. In contrast, high-IPC regimes enable the distilled dataset to support greater semantic diversity. Accordingly, we leverage a larger and more diverse set of teacher models to generate fine-grained soft label distributions, which better capture the intrinsic structure of the true label space. Formally, for each synthetic image, we assign the soft label as the averaged output from a set of IPC-dependent teachers:

$$\mathbf{t}(\mathbf{x}_i) = \frac{1}{|\mathbb{T}(|\mathcal{S}_{\mathbf{x}}|)|} \sum_{t \in \mathbb{T}(|\mathcal{S}_{\mathbf{x}}|)} F_t(\mathbf{x}_i) = \frac{1}{|\mathbb{T}(\text{IPC})|} \sum_{t \in \mathbb{T}(\text{IPC})} F_t(\mathbf{x}_i), \quad \text{for each } \mathbf{x}_i \in \mathcal{S}_{\mathbf{x}}, \tag{15}$$

where F_t denotes the logit output function of the t-th teacher, $\mathbf{t}(\mathbf{x}_i)$ denotes soft label for image $\mathbf{x}_i$, and $\mathbb{T}(\text{IPC})$ is a subset of teacher models selected to minimize the contraction factor α. This strategic relabeling ensures that the soft label distribution faithfully matches the capacity of the distilled images, reducing the discrepancy term $\mathrm{W}(\nu_{\text{distill}}^{(\text{soft})}, \mu_{\text{true}})$ and thereby improving alignment. For a fair comparison with prior methods [27, 28], we adopt the same region-level soft label storage strategy as in FKD [48].

4.5 OT-based Student Model Logit Matching (OTM)

After obtaining a distilled set that preserves rich geometric structures of the real data, we transfer this information to student models (i.e., new models) by aligning the distribution of their logits with that of the real dataset. We achieve this alignment by minimizing the last term in the upper bound of Equation 8. We consider a batch of b samples and denote the soft labels of this batch and the logit output of a student model as $\mathbf{t}$ and $\mathbf{s}$, respectively. Most traditional divergence measures operate on a per-sample basis and match logits independently, thereby failing to capture inter-sample relationships. To address this limitation, we employ a batch-wise OT distance that aligns logits while capturing global distributional structure. Specifically, similar to Section 4.3, we use the Sinkhorn method to efficiently solve for the OT matrix $\mathbf{P}^{\lambda_2}$, with entropy regularization $h(\mathbf{P})$ weighted by λ_2:

$$\mathbf{P}^{\lambda_2} = \underset{\mathbf{P}}{\operatorname{argmin}} \langle \mathbf{P}, \mathbf{C}(\mathbf{t}, \mathbf{s}) \rangle - \lambda_2 h(\mathbf{P}), \text{ where } \sum_i \mathbf{P}_{ij} = \frac{1}{b} \forall j, \ \sum_j \mathbf{P}_{ij} = \frac{1}{b} \forall i. \tag{16}$$

Here, $\mathbf{C} \in \mathbb{R}^{b \times b}$ is the batch-wise cost matrix, where each entry $\mathbf{C}_{ij}$ measures the distance between the soft label $\mathbf{t}(\mathbf{x}_i)$ and the synthetic output $\mathbf{s}(\mathbf{x}_j)$. Specifically, we employ the ℓ_p-norm:

$$\mathbf{C}_{ij}(\mathbf{t}, \mathbf{s}) = \| \mathbf{t}(\mathbf{x}_i) - \mathbf{s}(\mathbf{x}_j) \|_p, \quad \mathcal{L}_{\text{SD}} = \mathrm{W}(\mathbf{t}, \mathbf{s}) = \left\langle \mathbf{P}^{\lambda_2}, \mathbf{C} \right\rangle, \tag{17}$$

Adapting Equations 11, 12, and 13 to current dimensions, we compute $\mathbf{P}^{\lambda_2}$ and the corresponding batch-wise Sinkhorn distance loss $\mathcal{L}_{\text{SD}}$. For a fair comparison with previous methods [41, 23], we use the cross-entropy loss $\mathcal{L}_{\text{CE}}$, the MSE loss $\mathcal{L}_{\text{MSE}}$, and the Sinkhorn loss $\mathcal{L}_{\text{SD}}$ for distillation:

$$\mathcal{L} = \sum_{i=1}^{b} \kappa_1 \mathcal{L}_{\text{CE}}(y_{\text{onehot}}(\mathbf{x}_i), \mathbf{s}(\mathbf{x}_i)) + \kappa_2 \mathcal{L}_{\text{MSE}}(\mathbf{t}(\mathbf{x}_i), \mathbf{s}(\mathbf{x}_i)) + \beta_2 \mathcal{L}_{\text{SD}}, \tag{18}$$

where κ_1, κ_2, and β_2 are scalar weights, $y_{\text{onehot}}(\mathbf{x}_i)$ denotes the hard label for the distilled image $\mathbf{x}_i$.

Table 2: Performance comparison on ImageNet-1K [22] with ResNet-18. The numbers in parentheses for "Ours" represent the number of training epochs on the distilled set for new models.

Comparsion with generative-model-based methods.									
IPC	D^3M [31]	D^4M [32]	TDSDM [33]	DiT [44]	Minimax [45]	DDPS [46]	DiT-IGD [24]	Ours (300)	Ours (1000)
10	23.6±0.1	27.9±0.7	44.5±0.4	39.6±0.4	42.1±0.3	42.1±0.3	45.5±0.5	**52.9±0.1**	**58.6±0.3**
50	32.2±0.1	55.2±0.3	59.4±0.3	52.9±0.6	59.4±0.2	59.4±0.2	59.8±0.3	**61.9±0.5**	**64.2±0.4**
Comparsion with model-inversion-based methods									
IPC	SRe^2L [27]	G-VBSM [41]	RDED [28]	CDA [26]	SC-DD [42]	EDC [23]	CV-DD [25]	Ours (300)	Ours (1000)
10	21.3±0.6	31.4±0.5	42.0±0.1	33.5±0.3	32.1±0.2	48.6±0.3	46.0±0.6	**52.9±0.1**	**58.6±0.3**
50	46.8±0.2	51.8±0.4	56.5±0.1	53.5±0.3	53.1±0.1	58.0±0.2	49.5±0.4	**61.9±0.5**	**64.2±0.4**

Table 3: Other architecture performance comparison on ImageNet-1K [22].

Method	MobileNet-V2		EfficientNet-B0		Swin Transformer		ConvNeXt	
	IPC10	IPC50	IPC10	IPC50	IPC10	IPC50	IPC10	IPC50
SRe^2L [27]	10.2±2.6	31.8±0.3	11.4±2.5	34.8±0.4	4.8±0.6	42.1±0.3	4.1±0.4	48.8±0.2
RDED [28]	40.4±0.1	53.3±0.2	31.0±0.1	58.5±0.4	42.3±0.6	53.2±0.8	48.3±0.5	65.4±0.4
EDC [23]	45.0±0.2	57.8±0.1	51.1±0.3	60.9±0.2	46.0±0.5	57.9±0.3	54.4±0.2	66.6±0.2
DiT-IGD [44]	39.2±0.2	57.8±0.2	47.7±0.1	62.0±0.1	44.1±0.6	58.6±0.5	51.9±0.2	66.8±0.5
Ours (300)	**51.0±0.6**	**61.0±0.4**	**56.7±0.2**	**64.4±0.1**	**50.2±0.2**	**68.2±0.1**	**61.2±0.1**	**70.2±0.8**
Ours (500)	**54.6±0.3**	**63.0±0.4**	**59.6±0.2**	**66.0±0.6**	**56.2±1.0**	**69.4±0.1**	**64.5±0.3**	**71.1±1.1**
Ours (1000)	**57.6±0.1**	**63.9±0.2**	**62.4±0.1**	**66.8±0.1**	**63.7±0.2**	**70.5±0.1**	**67.0±0.1**	**71.8±0.9**

5 Experiments

5.1 Experimental Settings

Dataset. Given our primary focus on large-scale dataset distillation, we evaluate our method on the full ImageNet-1K dataset [22]. To ensure comparability across varying category scales, we further conduct experiments on two widely used subsets, ImageNet-100 [13] and ImageNette [49]. To construct a comprehensive benchmark covering both low-resolution and high-resolution settings, we additionally include CIFAR-100 [21]. The dataset descriptions are presented in Appendix C.

Network architectures. To evaluate the generalization capability of our method, we experiment with a diverse set of network architectures, including convolutional neural networks (CNNs), transformer-based models, and hybrid models. Specifically, we consider CNN-based architectures, including ResNet [50], MobileNet [51],EfficientNet [52], and ConvNet [53]; a transformer-based model, the Swin Transformer [54]; and the hybrid architecture ConvNeXt [55]. This selection provides a comprehensive evaluation across diverse architectural paradigms and inductive biases.

Baselines. We compare our approach with a broad range of dataset distillation methods. Specifically, we include traditional methods such as DM [16], IDC [13], and DATM [56]; model-inversion-based methods including SRe^2L [27], G-VBSM [41], RDED [28], CDA [26], SC-DD [42], EDC [23], CV-DD [25], and DELT[57]; as well as generative-model-based methods such as D^3M [31], D^4M [32], TDSDM [33], DiT [44], Minimax [45], DDPS [46], and IGD [24]. We report the top-1 test accuracy of models trained on distilled datasets with different IPC (Images Per Class) settings to ensure a fair and consistent comparison. Each network is trained five times from scratch to report error bars.

Implementation details. To ensure fair evaluation, we follow the configurations of IGD [24] and EDC [23], maintaining consistency in training procedure and hyperparameter settings. For the OT components, we set $\alpha_1 = 1$, $\gamma_1 \in \{1000, 3000\}$, $\alpha_2 = 0.1$, and $\gamma_2 = 0.1$. For simplicity, we set $p = 1$ (ℓ_1-norm). More details are in Appendix F.

5.2 Results and Discussions

Results on ImageNet-1K. We extensively evaluated our generative OT framework on ImageNet-1K [22], comparing it against state-of-the-art dataset distillation methods, including both generative model-based and model-inversion-based approaches, across various architectures and IPC settings. Table 2 presents results on ResNet-18 [50]. Our method significantly outperforms prior methods

Table 4: Performance comparison on ImageNette [49].

Model	ConvNet-6			ResNetAP-10			ResNet-18		
IPC	10	50	100	10	50	100	10	50	100
Hard Label									
Random	46.0±0.5	71.8±1.2	79.9±0.8	54.2±1.2	77.3±1.0	81.1±0.6	55.8±1.0	75.8±1.1	82.0±0.4
DM [16]	49.8±1.1	70.3±0.8	78.5±0.8	60.2±0.7	76.7±1.1	80.9±0.7	60.9±0.7	75.0±1.0	81.5±0.4
IDC-1 [13]	48.2±1.2	72.4±0.7	80.6±1.1	60.4±0.6	77.4±0.7	81.5±1.2	61.0±0.8	77.5±1.0	81.7±0.8
DiT [44]	56.2±1.3	74.1±0.6	78.2±0.3	62.8±0.8	76.9±0.5	80.1±1.1	62.5±0.9	75.2±0.7	77.8±0.7
Minimax [45]	58.2±0.9	76.9±0.8	81.1±0.3	63.2±1.0	78.2±0.7	81.5±1.0	64.9±0.6	78.1±0.6	81.3±0.7
DiT-IGD [44]	61.9±1.9	80.9±0.9	84.5±0.7	66.5±1.1	81.0±1.2	85.2±0.8	67.7±0.3	80.4±0.8	84.4±0.8
Ours	**67.0±0.9**	**83.1±1.0**	**86.5±0.5**	**68.0±0.3**	**83.8±0.6**	**86.4±0.6**	**69.1±1.9**	**84.6±0.4**	**85.9±0.2**
Soft Label									
SRe^{2}L [27]	-	-	-	-	-	-	29.4±3.0	40.9±0.3	50.2±0.4
RDED [28]	63.5±0.6	84.3±0.3	89.2±0.7	60.8±0.5	80.5±0.3	89.3±0.6	61.4±0.4	80.4±0.4	89.6±1.0
D^{4}M [32]	53.5±0.5	84.4±0.4	89.6±0.2	56.2±0.3	84.7±0.5	90.2±0.3	57.4±0.4	84.8±0.2	90.4±0.7
DDPS^c [46]	-	-	-	-	-	-	62.5±0.2	83.4±0.5	90.2±0.2
DDPS^s [46]	-	-	-	-	-	-	60.4±0.3	85.8±0.4	91.6±0.4
DiT-IGD* [24]	69.6±1.0	86.7±0.9	89.9±0.6	73.6±1.3	86.8±1.0	90.6±0.6	74.8±0.7	86.4±0.9	90.7±0.5
Ours	**74.5±0.3**	**89.1±0.9**	**91.3±0.2**	**77.8±0.8**	**89.7±0.5**	**91.6±0.3**	**79.0±0.3**	**89.3±0.3**	**92.0±0.6**
Full		94.3±0.5			94.6±0.5			95.3±0.6	

Table 5: Performance comparison on ImageNet-100 [13].

Model	IPC	SRe^{2}L [27]	RDED [28]	DELT [57]	Ours
ResNet-18	10	9.5±0.4	36.0±0.3	28.2±1.5	**47.7±0.3**
	50	27.0±0.4	61.6±0.1	67.9±0.6	**72.6±0.1**
	100	30.4±0.3	74.5±0.4	75.1±0.2	**79.2±0.1**
ResNet-101	10	6.4±0.1	33.9±0.1	22.4±3.3	**36.3±0.5**
	50	25.7±0.3	66.0±0.6	70.8±2.3	**74.3±0.2**
	100	27.6±0.2	73.5±0.8	77.6±1.8	**81.6±0.1**
MobileNet	10	4.5±0.4	23.6±0.7	15.8±0.2	**43.2±0.2**
	50	18.4±0.2	51.5±0.8	55.0±1.8	**69.5±0.3**
	100	22.1±0.3	70.8±1.1	76.7±0.3	**78.0±0.2**

Table 6: Performance comparison on CIFAR-100 [21] using ConvNet-3 [53].

IPC	10	50	100
DM [16]	29.7±0.3	43.6±0.4	47.1±0.4
M3D [18]	42.4±0.2	50.9±0.7	52.1±0.6
DATM [56]	47.2±0.4	55.0±0.2	57.5±0.2
SRe^{2}L [27]	24.5±0.4	45.2±0.3	46.6±0.5
RDED [28]	48.1±0.3	57.0±0.1	58.1±0.4
D^{4}M [32]	45.0±0.1	48.8±0.3	50.3±0.2
DiT-IDG [24]	45.8±0.6	53.9±0.6	55.9±0.4
Ours	**50.7±0.2**	**57.5±0.3**	**58.7±0.2**

at 300 epochs. When training is extended to 1000 epochs, performance further improves. This shows that our distilled images and soft labels contain sufficient information for continued optimization. Beyond ResNet, we evaluated generalization on MobileNet-V2 [51], EfficientNet-B0 [52], Swin Transformer [54], and ConvNeXt [55] (Table 3). Our framework consistently surpasses prior approaches across all architectures. The larger performance gain at lower IPC settings highlights its ability to better preserve fine-grained distributional details. When IPC is low, existing dataset distillation methods struggle to cover the full data distribution, leading to significant discrepancies between the learned distribution and the real distribution. In contrast, our approach explicitly aligns the latent space distribution, logit-level semantic consistency, and label-image relationships, ensuring that even with limited synthetic samples, our distilled set comprehensively represents the real data.

Results on ImageNet subsets. To further compare with prior works and to evaluate our method under reduced-category settings, we conduct experiments on two ImageNet subsets, varying both class selection and class count. As shown in Tables 4 and 5, our method consistently outperforms all baselines. Notably, we observe significant performance improvements under both hard label and soft label settings. This demonstrates that OT-guided sampling effectively captures fine-grained sample information, contributing to the learning of the new model. During the subsequent OT distance minimization phases, this extracted information is systematically transferred to the new model, resulting in enhanced performance. Robustness tests are conducted in Appendix G.8.

Results on CIFAR-100. We evaluate our method on CIFAR-100 [21] to assess its generalizability on low-resolution datasets, as summarized in Table 6. To ensure a broad comparison, we include traditional low-resolution-oriented methods, along with model-inversion-based and generative-model-based methods, both specially designed for large-scale datasets. Unlike most existing methods that specialize in either low-resolution or high-resolution datasets, our approach achieves state-of-the-art performance on ImageNet [22] while maintaining superior results on CIFAR. This further highlights the robustness of our OT-driven strategy in preserving distributional characteristics across scales.

Table 7: Ablation Study on ImageNette [49] under IPC=10. *Note*: Here, "w/o LIA" denotes soft relabeling with the teacher ensemble from high-IPC settings, without adapting to the current IPC.

Model	Hard Label		Soft Label			
	w/o OTG	w OTG	w/o OTG	w/o LIA	w/o OTM	Full
ConvNet-6	61.9	**67.0**	72.5	74.3	73.2	**74.5**
ResNetAP-10	66.5	**68.0**	74.2	76.4	75.9	**77.8**
ResNet-18	67.7	**69.1**	77.2	77.8	77.5	**79.0**

Table 8: Mean runtime per class (sampling) or per epoch (matching) on ImageNet-1K [22].

Stage	Method	IPC=10	IPC=50
Samp.	w/o OTG	97.1s	537.4s
	w OTG	97.7s	540.3s
Match.	w/o OTM	23.2s	126.1s
	w OTM	23.3s	126.6s

Table 9: Distilled set generation time (IPC=10, ImageNet-1K, 8×4090). PreS: Presample, PostS: Postsample.

EDC [23]	3h PreS +3h PostS + 5h Recover + 0.4h Relabel
Ours	3.4h Diffusion Sample + 0.3h Relabel

Table 10: Effect of α on ImageNette [49].

Teachers	ResNet-18	w/o LIA	w LIA
α	0.906	0.903	**0.643**
Avg. Acc.	76.0	76.2	**77.1**

Impact of different components. Our OT-guided diffusion sampling effectively transfers the geometric structure of the image space distribution to the distilled images. This alignment is further enhanced by the Label-Image-Aligned Soft Relabeling, which narrows the distributional gap between the distilled and real data. During student model training, the OT-based student logit matching module faithfully propagates this information to the new model. This further reinforces alignment between the original distribution μ_{true} and the learned distribution ν_{new}. As shown in Table 7, each component involved in minimizing the OT distance plays a critical role, underscoring the necessity of aligning distributions throughout the entire pipeline. More validations are provided in Appendix G.3.

Runtime analysis. As shown in Table 8, the additional time overhead introduced by our OT constraint is consistently less than 1%. Table 9 provides a breakdown of the time required for each step in generating the distilled set for both our method and the state-of-the-art model-inversion-based method, EDC [23]. Our approach is notably faster than EDC, which further demonstrates its efficiency.

Discussion of contraction factor α. Table 10 reports the values of α measured under different soft label generation strategies. Our LIA strategy significantly reduces the OT distance between the distilled data and the real data, allowing the distilled data to capture more information of the real distribution. This leads to a substantial performance improvement. More discussion in Appendix G.2.

Sensitivity analysis. As shown in Table 11, our method delivers consistently high accuracy over a broad range of OT-related hyperparameter settings, demonstrating low sensitivity. This robustness eliminates exhaustive tuning and enables straightforward deployment across diverse scenarios. It also makes our approach readily scalable. Additional results and analyses are available in Appendix G.1.

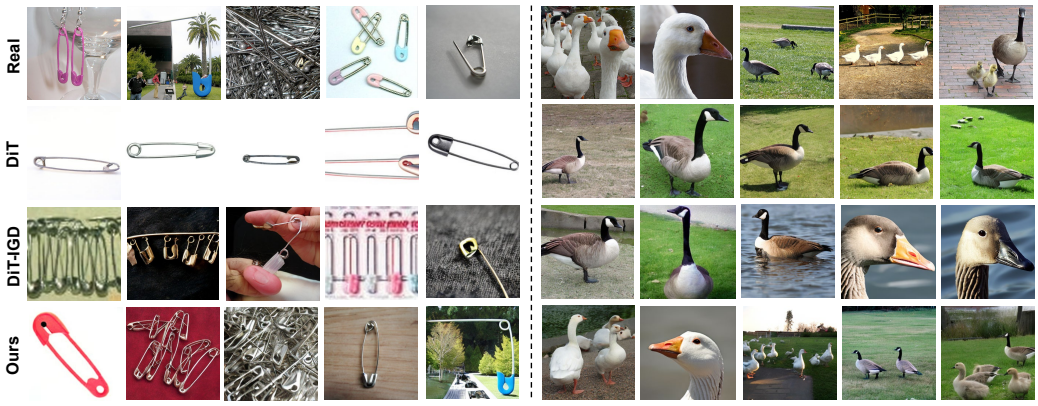


Figure 2: Comparison of generated images from different methods on ImageNet-100 (IPC = 10).

Table 11: Impact of different OT hyperparameters.

β_1	λ_1	β_2	λ_2	ConvNet	ResNetAP	ResNet
Hard Label						
1	1000	-	-	67.0±0.9	68.0±0.3	69.1±1.9
1	10000	-	-	66.3±0.7	68.5±0.3	68.9±0.8
10	1000	-	-	65.8±0.5	67.5±0.5	68.7±1.1
Soft Label						
1	1000	0.1	0.1	74.5±0.3	77.8±0.8	79.0±0.3
1	1000	0.1	1	74.6±0.8	76.3±0.8	78.2±1.0
1	1000	1	0.1	74.3±0.5	78.1±0.3	77.4±0.2

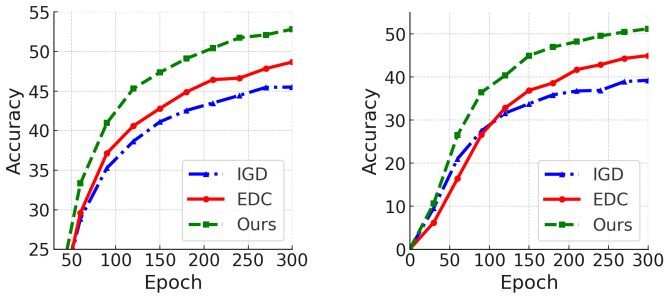


Figure 3: Training accuracy curves of ResNet-18 (left) and MobileNet (right). IGD: DiT-IGD.

Visualizations. Figure 2 illustrates a qualitative comparison among DiT [44], DiT-IGD [24], and our method. DiT often produces visually similar outputs that lack semantic diversity. DiT-IGD introduces diversity without aligning with the underlying real data distribution, leading to non-representative or incorrect generations. Furthermore, its influence estimation is based solely on intraclass averaged statistics, which results in perceptible blurring. In contrast, our approach explicitly models both instance-specific characteristics and fine-grained distributional structures, thereby enabling faithful approximation of the real data manifold. We also present the test accuracy at each logit-matching step in Figure 3. Our method achieves faster convergence and consistently higher accuracy, especially in early epochs, demonstrating superior sample informativeness and stronger distribution alignment when compared to EDC [23] and DiT-IGD [24]. Please refer to Appendix H for more visualizations.

6 Conclusion

We propose a principled framework for generative large-scale dataset distillation by formulating it as an OT distance minimization problem. Our approach explicitly decomposes the total OT distance into three interpretable components and systematically minimizes each to ensure comprehensive distributional alignment. This allows new models trained on distilled data to behave similarly to models trained on the full dataset, regardless of architecture. Extensive experiments across diverse datasets and model architectures validate the effectiveness and generalizability of our method.

Broader Impact. Our distilled datasets lower carbon footprints associated with new model training, fostering sustainable AI development. They also enable efficient learning in federated and continual learning scenarios, enhancing data privacy and model adaptation across distributed systems.

Acknowledgement This work was supported by the GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC, and the Supercomputing Center of the USTC.

Competing Interests The authors declare no competing interests.

References

[1] Ruonan Yu, Songhua Liu, and Xinchao Wang. Dataset distillation: A comprehensive review. *arXiv preprint arXiv:2301.07014*, 2023.

[2] Ping Liu and Jiawei Du. The evolution of dataset distillation: Toward scalable and generalizable solutions. *arXiv preprint arXiv:2502.05673*, 2025.

[3] Xiao Cui, Qi Sun, Min Wang, Li Li, Wengang Zhou, and Houqiang Li. Layoutenc: Leveraging enhanced layout representations for transformer-based complex scene synthesis. *ACM Transactions on Multimedia Computing, Communications and Applications*, 2025.

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

[5] Hongxu Yin, Pavlo Molchanov, Jose M. Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K. Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8715–8724, 2020.

[6] Yuqi Jia, Saeed Vahidian, Jingwei Sun, Jianyi Zhang, Vyacheslav Kungurtsev, Neil Zhenqiang Gong, and Yiran Chen. Unlocking the potential of federated learning: The symphony of dataset distillation via deep generative latents. In *European Conference on Computer Vision (ECCV)*, pages 18–33. Springer, 2024.

[7] Ze Chai, Zhipeng Gao, Yijing Lin, Chen Zhao, Xinlei Yu, and Zhiqiang Xie. Adaptive backdoor attacks against dataset distillation for federated learning. In *IEEE International Conference on Communications (ICC)*, pages 4614–4619, 2024.

[8] Enneng Yang, Li Shen, Zhenyi Wang, Tongliang Liu, and Guibing Guo. An efficient dataset condensation plugin and its application to continual learning. *Conference on Neural Information Processing Systems (NeurIPS)*, 36, 2023.

[9] Jianyang Gu, Kai Wang, Wei Jiang, and Yang You. Summarizing stream data for memory-constrained online continual learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pages 12217–12225, 2024.

[10] Lechao Cheng, Kaifeng Chen, Jiyang Li, Shengeng Tang, Shufei Zhang, and Meng Wang. Dataset distillers are good label denoisers in the wild. *arXiv preprint arXiv:2411.11924*, 2024.

[11] Dongyao Zhu, Bowen Lei, Jie Zhang, Yanbo Fang, Yiqun Xie, Ruqi Zhang, and Dongkuan Xu. Rethinking data distillation: Do not overlook calibration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4935–4945, 2023.

[12] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations (ICLR)*, 2020.

[13] 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. *arXiv preprint arXiv:2205.14959*, 2022.

[14] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4750–4759, 2022.

[15] Wenliang Zhong, Haoyu Tang, Qinghai Zheng, Mingzhu Xu, Yupeng Hu, and Liqiang Nie. Towards stable and storage-efficient dataset distillation: Matching convexified trajectory. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[16] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6514–6523, 2023.

[17] Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7856–7865, 2023.

[18] Hansong Zhang, Shikun Li, Pengju Wang, Dan Zeng, and Shiming Ge. M3d: Dataset condensation by minimizing maximum mean discrepancy. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pages 9314–9322, 2024.

[19] Shaobo Wang, Yicun Yang, Zhiyuan Liu, Chenghao Sun, Xuming Hu, Conghui He, and Linfeng Zhang. Dataset distillation with neural characteristic function: A minmax perspective. *arXiv preprint arXiv:2502.20653*, 2025.

[20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE (Proc. IEEE)*, 86(11):2278–2324, 1998.

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*, 2009.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 2015.

[23] Shitong Shao, Zikai Zhou, Huanran Chen, and Zhiqiang Shen. Elucidating the design space of dataset condensation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[24] Mingyang Chen, Jiawei Du, Bo Huang, Yi Wang, Xiaobo Zhang, and Wei Wang. Influence-guided diffusion for dataset distillation. In *International Conference on Learning Representations (ICLR)*, 2025.

[25] Jiacheng Cui, Zhaoyi Li, Xiaochen Ma, Xinyue Bi, Yaxin Luo, and Zhiqiang Shen. Dataset distillation via committee voting. *arXiv preprint arXiv:2501.07575*, 2025.

[26] Zeyuan Yin and Zhiqiang Shen. Dataset distillation in large data era. *Transactions on Machine Learning Research (TMLR)*, 2024.

[27] Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[28] Peng Sun, Bei Shi, Daiwei Yu, and Tao Lin. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[29] Hongxu Yin, Pavlo Molchanov, Jose M. Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K. Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8715–8724, 2020.

[30] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. In *IEEE International Conference on Computer Vision (ICCV)*, pages 9374–9384, 2021.

[31] Ali Abbasi, Ashkan Shahbazi, Hamed Pirsiavash, and Soheil Kolouri. One category one prompt: Dataset distillation using diffusion models. *arXiv preprint arXiv:2403.07142*, 2024.

[32] Duo Su, Junjie Hou, Weizhi Gao, Yingjie Tian, and Bowen Tang. Dˆ4: Dataset distillation via disentangled diffusion model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[33] Jianhao Yuan, Jie Zhang, Shuyang Sun, Philip Torr, and Bo Zhao. Real-fake: Effective training data synthesis through distribution matching. *arXiv preprint arXiv:2310.10402*, 2023.

[34] Xiao Cui, Mo Zhu, Yulei Qin, Liang Xie, Wengang Zhou, and Houqiang Li. Multi-level optimal transport for universal cross-tokenizer knowledge distillation on language models. *AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

[35] Xiao Cui, Yulei Qin, Yuting Gao, Enwei Zhang, Zihan Xu, Tong Wu, Ke Li, Xing Sun, Wengang Zhou, and Houqiang Li. Sinkd: Sinkhorn distance minimization for knowledge distillation. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2024.

[36] Yanqing Liu, Jianyang Gu, Kai Wang, Zheng Zhu, Wei Jiang, and Yang You. Dream: Efficient dataset distillation by representative matching. In *IEEE International Conference on Computer Vision (ICCV)*, pages 17314–17324, 2023.

[37] Hansong Zhang, Shikun Li, Fanzhao Lin, Weiping Wang, Zhenxing Qian, and Shiming Ge. Dance: Dual-view distribution alignment for dataset condensation. *arXiv preprint arXiv:2406.01063*, 2024.

[38] Xiao Cui, Yulei Qin, Wengang Zhou, Hongsheng Li, and Houqiang Li. Optical: Leveraging optimal transport for contribution allocation in dataset distillation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[39] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Generalizing dataset distillation via deep generative prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[40] Jun-Yeong Moon, Jung Uk Kim, and Gyeong-Moon Park. Towards model-agnostic dataset condensation by heterogeneous models. arXiv preprint arXiv:2409.14538, 2024.

[41] Shitong Shao, Zeyuan Yin, Muxin Zhou, Xindong Zhang, and Zhiqiang Shen. Generalized large-scale data condensation via various backbone and statistical matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[42] Muxin Zhou, Zeyuan Yin, Shitong Shao, and Zhiqiang Shen. Self-supervised dataset distillation: A good compression is all you need. *arXiv preprint arXiv:2404.07976*, 2024.

[43] Haoyang Liu, Yuchen Li, Tianyu Xing, Vivek Dalal, Lirong Li, Jun He, and Hao Wang. Dataset distillation via the wasserstein metric. *arXiv preprint arXiv:2311.18531*, 2023.

[44] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4195–4205, 2023.

[45] Jianyang Gu, Saeed Vahidian, Vyacheslav Kungurtsev, Haonan Wang, Wei Jiang, Yang You, and Yiran Chen. Efficient dataset distillation via minimax diffusion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[46] Xinhao Zhong, Shuoyang Sun, Xulin Gu, Zhaoyang Xu, Yaowei Wang, Min Zhang, and Bin Chen. Efficient dataset distillation via diffusion-driven patch selection for improved generalization. *arXiv preprint arXiv:2412.09959*, 2024.

[47] Jiwen Yu, Wei Wang, Xiaobo Zhang, Bo Huang, and Yi Wang. Freedom: Training-free energy-guided conditional diffusion model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4567–4576, 2023.

[48] Zhiqiang Shen and Eric Xing. A fast knowledge distillation framework for visual recognition. *arXiv preprint arXiv:2112.01528*, 2021.

[49] Jeremy Howard. A smaller subset of 10 easily classified classes from imagenet, and a little more french. *Project URL: https://github.com/fastai/imagenette*, 4, 2019.

[50] Kaiming He, Xiangyu Zhang, and Shaoqing Ren. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, Jun. 2016. IEEE.

[51] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, Salt Lake City, UT, USA, Jun. 2018. IEEE.

[52] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, pages 6105–6114. PMLR, 2019.

[53] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[54] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021.

[55] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11976–11986, 2022.

[56] Ziyao Guo, Kai Wang, George Cazenavette, Hui Li, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching. In *International Conference on Learning Representations (ICLR)*, 2024.

[57] Zhiqiang Shen, Ammar Sherif, Zeyuan Yin, and Shitong Shao. Delt: A simple diversity-driven earlylate training for dataset distillation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[58] Cédric Villani and Cédric Villani. The wasserstein distances. *Monograph: Optimal Transport: Old and New*, pages 93–111, 2009.

[59] Jingwei Zhang, Tongliang Liu, and Dacheng Tao. An optimal transport analysis on generalization in deep learning. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 34(6):2842–2853, 2021.

[60] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning (ICML)*, pages 214–223, 2017.

[61] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. *Conference on Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[62] Artan Sheshmani, Yi-Zhuang You, Baturalp Buyukates, Amir Ziashahabi, and Salman Avestimehr. Renormalization group flow, optimal transport, and diffusion-based generative model. *Physical Review E (Phys. Rev. E)*, 111(1):015304, 2025.

[63] Dylan Wheeler and Balasubramaniam Natarajan. Conceptual learning and causal reasoning for semantic communication. *IEEE Transactions on Cognitive Communications and Networking (TCCN)*, 2025.

[64] Weilin Chen, Jie Qiao, Ruichu Cai, and Zhifeng Hao. On the role of entropy-based loss for learning causal structure with continuous optimization. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2023.

[65] Tung Le, Khai Nguyen, Shanlin Sun, Nhat Ho, and Xiaohui Xie. Integrating efficient optimal transport and functional maps for unsupervised shape correspondence learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23188–23198, 2024.

[66] Yonghao Liu, Fausto Giunchiglia, Ximing Li, Lan Huang, Xiaoyue Feng, and Renchu Guan. Enhancing unsupervised graph few-shot learning via set functions and optimal transport. *arXiv preprint arXiv:2501.05635*, 2025.

[67] Eduardo Fernandes Montesuma, Adel El Habazi, and Fred Ngole Mboula. Unsupervised anomaly detection through mass repulsing optimal transport. *arXiv preprint arXiv:2502.12793*, 2025.

[68] Ali Baheri, Zahra Sharooei, and Chirayu Salgarkar. Wasserstein adaptive value estimation for actor-critic reinforcement learning. *arXiv preprint arXiv:2501.10605*, 2025.

[69] Pascal Klink, Carlo D'Eramo, Jan Peters, and Joni Pajarinen. On the benefit of optimal transport for curriculum reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2024.

[70] Yixing Lan, Xin Xu, Qiang Fang, and Jianye Hao. Sample efficient deep reinforcement learning with online state abstraction and causal transformer model prediction. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2023.

[71] Qiyuan Zhang, Shu Leng, Xiaoteng Ma, Qihan Liu, Xueqian Wang, Bin Liang, Yu Liu, and Jun Yang. Cvar-constrained policy optimization for safe reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2024.

[72] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Conference on Neural Information Processing Systems (NeurIPS)*, 26, 2013.

[73] Jiafei Lyu, Mengbei Yan, Zhongjian Qiao, Runze Liu, Xiaoteng Ma, Deheng Ye, Jing-Wen Yang, Zongqing Lu, and Xiu Li. Cross-domain offline policy adaptation with optimal transport and dataset constraint. In *International Conference on Learning Representations (ICLR)*, 2025.

[74] Zhichen Zeng, Boxin Du, Si Zhang, Yinglong Xia, Zhining Liu, and Hanghang Tong. Hierarchical multi-marginal optimal transport for network alignment. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pages 16660–16668, 2024.

[75] Okan Koç, Alexander Soen, Chao-Kai Chiang, and Masashi Sugiyama. Domain adaptation and entanglement: an optimal transport perspective. *arXiv preprint arXiv:2503.08155*, 2025.

[76] Khai Nguyen, Hai Nguyen, Tuan Pham, and Nhat Ho. Lightspeed geometric dataset distance via sliced optimal transport. *arXiv preprint arXiv:2501.18901*, 2025.

[77] Lian-Bao Jin, Na Lei, Zhong-Xuan Luo, Jin Wu, Chao Ai, and Xianfeng Gu. Semi-discrete optimal transport for long-tailed classification. *Journal of Computer Science and Technology (JCST)*, 40(1):252–266, 2025.

[78] Xiao Cui, Yulei Qin, Yuting Gao, Enwei Zhang, Zihan Xu, Tong Wu, Ke Li, Xing Sun, Wengang Zhou, and Houqiang Li. Sinkhorn distance minimization for knowledge distillation. In *International Joint Conference on Language Resources and Evaluation and International Conference on Computational Linguistics (LREC-COLING)*, pages 14846–14858, 2024.

[79] Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

[80] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pages 8780–8794, 2021.

[81] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[82] Tuo Wang, Jiaming Song, Michael Elad, and Stefano Ermon. Zero-shot image restoration using denoising diffusion restoration models. *arXiv preprint arXiv:2201.11793*, 2022.

[83] Bahjat Kawar, Jiaming Song, Michael Elad, and Stefano Ermon. Denoising diffusion restoration models. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 35, pages 23502–23516, 2022.

[84] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 35, pages 27439–27452, 2022.

[85] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 35, pages 14715–14728, 2022.

[86] Nithin Gopalakrishnan Nair and Vishal M. Patel. Steered diffusion: A generalized framework for plug-and-play conditional image generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12345–12354, 2023.

[87] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 123–132, 2023.

[88] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[89] Jiawei Du, Xin Zhang, Juncheng Hu, Wenxin Huang, and Joey Tianyi Zhou. Diversity-driven synthesis: Enhancing dataset distillation through directed weight adjustment. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[90] Yifan Wu, Jiawei Du, Ping Liu, Yuewei Lin, Wei Xu, and Wenqing Cheng. Dd-robustbench: An adversarial robustness benchmark for dataset distillation. *IEEE Transactions on Image Processing (TIP)*, 2025.

[91] Hoki Kim. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*, 2020.

[92] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Learning Representations (ICLR)*, pages 7472–7482, 2019.

[93] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Robustness may be at odds with accuracy. In *International Conference on Learning Representations (ICLR)*, 2019.

[94] Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. *Conference on Neural Information Processing Systems (NeurIPS)*, 31, 2018.

[95] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, pages 7472–7482. PMLR, 2019.

[96] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9(11):2579–2605, 2008.

[97] Xiao Cui, Wengang Zhou, Yang Hu, Weilun Wang, and Houqiang Li. Heredity-aware child face image generation with latent space disentanglement. *arXiv preprint arXiv:2108.11080*, 2021.

Appendix

A Overview

This appendix provides comprehensive supplementary materials to further elaborate on our method's theoretical foundations, experimental setup, and empirical findings. It includes the following sections:

- **Section B: More Related Work.** Detailed discussions on prior studies, with an emphasis on optimal transport and guided diffusion sampling
- **Section C: Dataset Descriptions.** Comprehensive descriptions of all datasets used in our experiments, including ImageNet-1K [22], ImageNette [49], ImageNet-100 [13], and CIFAR-100 [21].
- **Section D: Symbol Table.** A complete summary of key mathematical notations, hyperparameters, and definitions referenced throughout the paper.
- **Section E: Pseudocode.** Step-by-step pseudocode for the proposed pipeline, detailed procedures for OT-alignment in different stages, and the calculation process for the contraction factor α.
- **Section F: Implementation Details.** Full specifications of hyperparameter settings, training schedules, and augmentation strategies across all datasets used in our experiments.
- **Section G: Further Experimental Analyses.** Additional experiments, including sensitivity studies (G.1), in-depth analysis of the contraction factor (G.2), comparisons with alternative distance metrics (G.3), runtime analysis (G.4), data coverage evaluation (G.5), expanded comparisons with other baselines (G.6 and G.7), robustness evaluation under adversarial attacks (G.8) and evaluation under extremely low-IPC settings G.9.
  This section constitutes the core of the appendix, offering deep empirical analyses of the contraction factor α and the robustness properties of the distilled models. Other experiments further strengthen and extend the key findings presented in the main text.
- **Section H: More Visualization Results.** Additional qualitative visualizations, including t-SNE plots and synthesized images, to assess semantic coverage and distributional diversity.
- **Section I: Limitations.** Critical discussion of the imitations of our framework.
- **Section J: Broader Impact.** Reflections on the broader societal, ethical, and practical implications of our dataset distillation method.

Together, these supplementary materials provide a complete and transparent view of our method, support full reproducibility, and offer additional insights that complement and strengthen the main paper.

B More Related Work

Optimal transport OT theory provides a principled mathematical framework for comparing probability distributions by computing the minimal cost required to transform one distribution into another. Compared to KL divergence and Jensen-Shannon (JS) divergence, OT provides a more geometrically faithful measure of distributional differences, particularly when dealing with distributions with non-overlapping supports [58, 59]. The Wasserstein distance, also known as OT distance, effectively quantifies distributional discrepancies and has been widely applied in image generation [60, 61, 62], causal discovery [63, 64], unsupervised learning [65, 66, 67], and reinforcement learning [68, 69, 70, 71]. However, its exact computation is intractable for high-dimensional data due to prohibitive complexity. To overcome this, the Sinkhorn distance introduces entropy regularization, making OT computation more efficient and numerically stable [72]. This regularized variant extends OT applications to domain adaptation [73, 74, 75],classification [76, 77], and knowledge distillation [34, 35, 78]. In this work, we propose a generative model-based OT framework designed to achieve precise distributional alignment throughout the dataset distillation process. Our approach optimizes the distilled dataset to minimize the OT distance between any student model's output distribution and the real data distribution, ensuring improved generalization.

Guided diffusion sampling Guided diffusion sampling enhances the generative capabilities of pre-trained diffusion models by incorporating external guidance during the reverse process to steer generation toward desired semantics [79]. Early methods, such as classifier guidance [80], inject gradients from pre-trained classifiers into the sampling process to condition generation. However, this approach necessitates domain-specific classifiers trained on noisy intermediate latents, which are often impractical. Classifier-free guidance [81] addresses this limitation by training the model with both conditional and unconditional objectives, enabling control without external models. Building upon this, Wang et al. [82] introduced linear operator-based guidance, constraining the diffusion process to the null space of known measurement operators; nonetheless, this strategy faces challenges in generalizing to nonlinear mappings. Subsequent works [83, 84, 85] extend guidance to inverse problems through iterative optimization and plug-and-play conditioning. Concurrently, methods like those proposed by Gopalakrishnan Nair et al. [86], Yu et al. [47], and Bansal et al. [87] introduce generic guidance functions by injecting gradients from task-specific losses computed on denoised intermediate states, thereby broadening applicability without necessitating model retraining. Inspired by these methods, Influence-Guided Diffusion (IGD) [24] leverages guided diffusion for dataset distillation by modifying the reverse sampling process to generate training-optimal data. However, its reliance on matching global distributional trajectories and introducing diversity through random perturbations often leads to suboptimal alignment, neglecting discriminative yet informative local characteristics in favor of global averaging. To overcome this limitation, we propose an optimal transport-based guidance strategy that explicitly aligns the geometric structure of real and synthetic distributions, achieving fine-grained consistency in guided diffusion sampling.

C Dataset Description

ImageNet-1K ImageNet-1K [22], also known as the ILSVRC 2012 dataset, is a large-scale image classification benchmark comprising 1,000 object categories. It contains approximately 1.28 million training images, 50,000 validation images, and 100,000 test images. The dataset is organized according to the WordNet hierarchy, with each synset corresponding to a distinct semantic concept. ImageNet-1K has been instrumental in advancing deep learning research and remains a standard benchmark for evaluating image classification models in large-scale settings.

ImageNette ImageNette [49] is a curated subset of ImageNet, consisting of 10 relatively easy categories, including “tench”, “English springer”, “cassette player”, “chain saw”, “church”, “French horn”, “garbage truck”, “gas pump”, “golf ball”, and “parachute”. It was introduced to facilitate rapid experimentation and prototyping of image classification models, particularly under limited computational budgets. All images are resized to a resolution of 224×224 pixels, providing a lightweight yet meaningful benchmark for distillation and robustness studies.

ImageNet-100 ImageNet-100 [13] is another subset derived from ImageNet-1K, comprising 100 randomly selected classes. Each class typically contains around 1,000 training images and 300 test images, maintaining a relatively balanced distribution. ImageNet-100 provides a manageable yet challenging benchmark for evaluating classification performance, especially in scenarios where computational efficiency and rapid iteration are prioritized.

CIFAR-100 CIFAR-100 [21] is a widely used benchmark dataset for image classification, extending the number of classes from 10 (CIFAR-10) to 100. Each class contains 600 images, with all images having a resolution of 32×32 pixels. Despite its compact size, CIFAR-100 presents a significant classification challenge due to its high intra-class variability and fine-grained label structure, making it a valuable resource for developing and assessing lightweight classification models.

D Symbol Description

To enhance clarity, a detailed description of mathematic symbols in the present study is provided in Table 12.

Table 12: Descriptions of all symbols, functions, and hyperparameters introduced in the main paper.

Symbol	Definition
E	Encoder to transform image into the latent space
D	Decoder to reconstruct latent code back to the image space
$\mathcal{S}$	Distilled (synthetic) dataset
$\mathcal{T}$	Real (full) dataset
$\mathbf{x}$	Image
$\mathbf{z}_0$	Latent code of clean sample
$\mathbf{z}_t$	Latent code of noisy sample at time step t
$\mathbf{z}_t^c$	Latent code of class-c noisy sample at time step t
α_t	Noise schedule controlling the perturbation at time step t
$\boldsymbol{\epsilon}$	Gaussian noise
ϵ_ϕ	Denoising function parameterized by ϕ
s	Reverse diffusion update function
$\mathcal{G}$	Guidance function in guided diffusion
$\mathcal{G}_I$	Influence function for general alignment
$\mathcal{G}_D$	Diversity function enforcing diversity in distilled data
$\mathcal{G}_{\mathrm{W}}$	Guidance function based on optimal transport
$\mathbf{M}_n^c$	Previously sampled n latents for class c
$\hat{\mathbf{M}}_n^c$	Concatenation of $\mathbf{M}_{n-1}^c$ and the latent $\mathbf{z}_t^c$ under sampling
$\mathbf{Z}_{\mathcal{T}}^c$	Latent representations of class c from the real dataset
p	Norm order
$\mathbf{P}^{\lambda_1}$	Optimal transport matrix for guided diffusion sampling with regularization
$\mathbf{D}$	Latent space cost matrix for guided diffusion sampling
$\mathbf{K}^t$	Transport matrix at the t-th step of Sinkhorn normalization
T	Sinkhorn iterations
T_D	Diffusion denoising iterations
$\mathbf{P}^{\lambda_2}$	Optimal transport matrix for logit matching with regularization
$\mathbf{C}$	Batch-wise cost matrix for logit matching
F_t	The logit output function of the t-th teacher
$\mathbf{t}$	Soft label for a batch
$\mathbf{s}$	Student model output for a batch
$\mathbf{t}_i$	Soft label for the i-th image in a batch
$\mathbf{y}_{\text{onehot}}(\mathbf{x}_i)$	One-hot hard label for the i-th image in a batch
b	Batch size
$h(\mathbf{P})$	The entropy of $\mathbf{P}$
ν_{distill}	Distilled data distribution
$\nu_{\text{distill}}^{\text{soft}}$	Distilled data distribution with soft label
$\nu_{\text{distill}}^{\text{hard}}$	Distilled data distribution with hard label
μ_{true}	Real dataset distribution
ν_{new}	Output of the student model after training on the distilled set
$\mathrm{W}(\mu_{\text{true}}, \nu_{\text{new}})$	Wasserstein distance between real dataset and student model output
ρ_t	Weight for influence function in the reverse sampling process
γ_t	Weight for diversity function in the reverse sampling process
β_1	Weight for the optimal transport guidance in the reverse sampling process
λ_1	Entropy regularization weight for optimal transport matrix
λ_2	Entropy regularization weight for logit matching
$\alpha(\nu_{\text{distill}}^{(\text{soft})})$	Contraction factor quantifying the benefit of soft labels
IPC	Images per class in the dataset
$\mathbb{T}$	Set of teacher models
$\mathcal{S}_{\mathbf{x}}$	Distilled image set
κ_1	Weight for cross-entropy loss in logit matching
κ_2	Weight for mean squared error loss in logit matching
β_2	Weight for Sinkhorn distance loss in logit matching
$\mathcal{L}_{\text{CE}}$	Cross-entropy loss
$\mathcal{L}_{\text{MSE}}$	Mean squared error loss
$\mathcal{L}_{\text{SD}}$	Sinkhorn distance loss

E PseudoCode

We present the pseudocode for our pipeline in Algorithm 1. The detailed calculation of the optimal transport (OT) distance for OT-guided Diffusion Sampling is provided in Algorithm 2, while the OT-based Student Model Logit matching is outlined in Algorithm 3. For efficient computation, we approximate the contraction factors using features in the latent space, enabling dimensionality reduction while preserving critical information.

Algorithm 1 OT-based Generative Dataset Distillation Framework

Require: Real dataset $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}$, teacher models $\mathbb{T}$, target IPC, diffusion model G, encoder E, decoder D, student model S
Ensure: Distilled dataset $\mathcal{S}_\mathbf{x}$ and trained student model S

for each class $c = 1$ to C **do**
  Encode real samples: $\mathbf{Z}_\mathcal{T}^c \leftarrow E(\{\mathbf{x}_i : y_i = c\})$
  for sample index $n = 1$ to IPC **do**
    Sample latent $\mathbf{z}_{T_D}^c$ using diffusion model G
    for $t = T_D$ to 1 **do**
      Compute OT-guidance $\mathcal{G}_\mathrm{W}(\mathbf{z}_t^c)$ w.r.t. $\mathbf{Z}_\mathcal{T}^c$ and previously sampled latents $\mathbf{M}_{n-1}^c$
      Update latent using guidance: $\mathbf{z}_{t-1}^c \leftarrow s(\mathbf{z}_t^c, t, \epsilon_\phi) - \rho_t \nabla \mathcal{G}_I - \gamma_t \nabla \mathcal{G}_D - \beta_1 \nabla \mathcal{G}_\mathrm{W}$
    end for
    Append $\mathbf{z}_{t-1}^c$ to $\mathbf{M}^c$
  end for
end for
Decode all latents: $\mathcal{S}_\mathbf{x} \leftarrow D(\mathbf{M}_\mathrm{IPC}^c)$ for all c
Select teacher set $\mathbb{T}(\mathrm{IPC})$ according to IPC level
for each image $\mathbf{x}_i \in \mathcal{S}_\mathbf{x}$ **do**
  Generate soft label: $\mathbf{t}(\mathbf{x}_i) \leftarrow \frac{1}{|\mathbb{T}|} \sum_{t \in \mathbb{T}} F_t(\mathbf{x}_i)$
end for
for each training batch $\mathcal{B} \subset \mathcal{S}_\mathbf{x}$ **do**
  Get soft labels $\mathbf{t}$ and student outputs $\mathbf{s} \leftarrow S(\mathcal{B})$
  Compute batch-wise OT loss $\mathcal{L}_\mathrm{SD} \leftarrow \mathrm{W}(\mathbf{t}, \mathbf{s})$
  Compute per-sample CE and MSE loss: $\mathcal{L}_\mathrm{CE} = \sum \mathcal{L}_\mathrm{CE}(y_\mathrm{onehot}, \mathbf{s}), \quad \mathcal{L}_\mathrm{MSE} = \sum \mathcal{L}_\mathrm{MSE}(\mathbf{t}, \mathbf{s})$
  Total loss: $\mathcal{L} = \kappa_1 \mathcal{L}_\mathrm{CE} + \kappa_2 \mathcal{L}_\mathrm{MSE} + \beta_2 \mathcal{L}_\mathrm{SD}$
  Update student model S using gradient descent
end for
return $\mathcal{S}_\mathbf{x}$, S

Algorithm 2 Computation of OT-based Guidance for Image Latent Sampling

Require: Previously sampled latents $\mathbf{M}_{n-1}^c$, current latent $\mathbf{z}_t^c$, a random batch of real class latents $\mathbf{Z}_\mathcal{T}^c$, regularization weight λ_1, iteration number T
Ensure: Optimal transport distance as guidance value $\mathcal{G}_\mathrm{W}(\mathbf{z}_t^c)$

Concatenate latent: $\hat{\mathbf{M}}_n^c \leftarrow [\mathbf{M}_{n-1}^c, \mathbf{z}_t^c]$
Compute cost matrix: $\mathbf{D}_{ij} \leftarrow \|\hat{\mathbf{M}}_n^c(i) - \mathbf{Z}_\mathcal{T}^c(j)\|_p$
Initialize kernel matrix: $\mathbf{K} \leftarrow \exp(-\mathbf{D}/\lambda_1)$
Set iteration counter $t \leftarrow 0$
while $t < T$ **do**
  Row normalization: $\mathbf{K} \leftarrow \mathrm{diag}\left(\mathbf{K}\mathbf{1}_n \oslash (n \cdot \mathbf{1}_{|\mathbf{Z}_\mathcal{T}^c|})\right)^{-1} \cdot \mathbf{K}$
  Column normalization: $\mathbf{K} \leftarrow \mathbf{K} \cdot \mathrm{diag}\left(\mathbf{K}^\top \mathbf{1}_{|\mathbf{Z}_\mathcal{T}^c|} \oslash (|\mathbf{Z}_\mathcal{T}^c| \cdot \mathbf{1}_n)\right)^{-1}$
  Increment iteration counter $t \leftarrow t + 1$
end while
Final transport matrix: $\mathbf{P}^{\lambda_1} \leftarrow \mathbf{K}$
Compute guidance: $\mathcal{G}_\mathrm{W}(\mathbf{z}_t^c) \leftarrow \langle \mathbf{P}^{\lambda_1}, \mathbf{D} \rangle = \sum_{i,j} \mathbf{P}_{ij}^{\lambda_1} \mathbf{D}_{ij}$
return $\mathcal{G}_\mathrm{W}(\mathbf{z}_t^c)$

Algorithm 3 Computation of Batch-wise OT for Student Logit Matching

Require: Teacher output $\mathbf{t}$, Student output $\mathbf{s}$,
Hyper-parameter λ_2, Maximum number of iterations T
Ensure: Sinkhorn loss $\mathcal{L}_{\text{SD}}$
Apply softmax: $\mathbf{t} \leftarrow \mathrm{Softmax}(\mathbf{t}), \quad \mathbf{s} \leftarrow \mathrm{Softmax}(\mathbf{s})$
Compute distance matrix $\mathbf{C}_{ij}(\mathbf{t}, \mathbf{s}) = \parallel \mathbf{t}(\mathbf{x}_i) - \mathbf{s}(\mathbf{x}_j) \parallel_p$
Compute kernel matrix $\mathbf{K} \leftarrow \exp\left(-\frac{\mathbf{C}}{\lambda_2}\right)$
Set iteration counter $t \leftarrow 0$
while $t < T$ **do**
  Row normalization: $\mathbf{K} \leftarrow \mathbf{K} \oslash \left(\mathbf{K}\mathbf{1}_b\mathbf{1}_b^{\mathrm{T}}\right)$
  Column normalization: $\mathbf{K} \leftarrow \mathbf{K} \oslash \left(\mathbf{1}_b\mathbf{1}_b^{\mathrm{T}}\mathbf{K}\right)$
  Increment iteration counter $t \leftarrow t + 1$
end while
Sinkhorn loss $\mathcal{L}_{\text{SD}} \leftarrow \langle \mathbf{K}, \mathbf{C} \rangle = \sum_{i,j} \mathbf{K}_{ij}\mathbf{C}_{ij}$
return $\mathcal{L}_{\text{SD}}$

Algorithm 4 Class-wise OT Distance in Label–Image Space for Contraction Factor α Calculation

Require: Real latent sets $\mathbf{Z}_{\mathcal{T}} \in \mathbb{R}^{N_1 \times d}$, distilled latent sets $\mathbf{Z}_{\mathcal{S}} \in \mathbb{R}^{N_2 \times d}$;
One-hot labels $\mathbf{H}_{\mathcal{T}} \in \{0,1\}^{N_1 \times C}$, soft labels $\mathbf{S}_{\mathcal{S}} \in [0,1]^{N_2 \times C}$;
Regularization parameter $\varepsilon > 0$, iterations T
Ensure: Average classwise OT distance $\mathcal{L}_{\text{avg}}$
Compute pairwise cost matrix $\mathbf{C}_{ij} \leftarrow \|\mathbf{Z}_{\mathcal{T}}(i) - \mathbf{Z}_{\mathcal{S}}(j)\|_p$
Initialize list of valid class distances W
for each class $c = 1$ to C **do**
  $\tilde{\mathbf{a}} \leftarrow \mathbf{H}_{\mathcal{T}}[:, c], \quad \tilde{\mathbf{b}} \leftarrow \mathbf{S}_{\mathcal{S}}[:, c]$
  if $\sum_i \tilde{\mathbf{a}}_i = 0$ **or** $\sum_j \tilde{\mathbf{b}}_j = 0$ **then**
    continue
  end if
  $\mathbf{a} \leftarrow \tilde{\mathbf{a}} / \sum_i \tilde{\mathbf{a}}_i$
  $\mathbf{b} \leftarrow \tilde{\mathbf{b}} / \sum_j \tilde{\mathbf{b}}_j$
  $\mathbf{K} \leftarrow \exp(-\mathbf{C}/\varepsilon)$
  Initialize $\mathbf{u} \leftarrow \mathbf{1}/N_1$
  Initialize $\mathbf{v} \leftarrow \mathbf{1}/N_2$
  for $t = 1$ to T **do**
    $\mathbf{u} \leftarrow \mathbf{a}/(\mathbf{K} \cdot \mathbf{v} + \delta)$
    $\mathbf{v} \leftarrow \mathbf{b}/(\mathbf{K}^\top \cdot \mathbf{u} + \delta)$
  end for
  $\gamma \leftarrow \mathrm{diag}(\mathbf{u}) \cdot \mathbf{K} \cdot \mathrm{diag}(\mathbf{v})$
  $\mathcal{L}_c \leftarrow \sum_{i,j} \gamma_{ij}\mathbf{C}_{ij}$
  Append $\mathcal{L}_c$ to list $\mathcal{L}$
end for
W $\leftarrow \frac{1}{|\mathcal{L}|} \sum_c \mathcal{L}_c$
return W

F Implementation Details

To ensure a fair and rigorous evaluation, we adopt the training protocols and experimental configurations established by IGD [24] and EDC [23], maintaining full consistency in model architecture, optimization settings, and evaluation pipelines. Following Minimax [45] and IGD, we utilize a latent DiT model from Pytorch's official repository and an open-source VAE model from Stable Diffusion. DDIM [88] with 50 denoised steps is used as the vanilla sampling method for generation. Also, all hyperparameters related to trajectory and diversity guidance are directly inherited from IGD, while the settings for student model logit matching follow those of EDC, with the exception of parameters introduced by our optimal transport (OT) framework. Notably, most of the OT-specific hyperparameters are set to fixed values across all datasets, and we observe that they require minimal tuning to achieve strong performance. This demonstrates the robustness of our method and its low sensitivity to OT parameter variations. Comprehensive hyperparameter configurations for all benchmark datasets including ImageNet-1K, ImageNette, ImageNet-100, and CIFAR-100 are detailed in Tables 13, 14, 15, and 16, respectively.

Table 13: Hyperparameter setting on ImageNet-1K [22].

Config	**Value**	**Explanation**
	Guided Diffusion Sampling	
k	5	$\rho_t = k \cdot \sqrt{1-\alpha_t} \cdot \frac{\|\epsilon_\phi(\mathbf{z}_t, t, c)\|}{\left\|\nabla_{\mathbf{z}_t} \mathcal{G}_I(\hat{\mathbf{z}}_0 \mid t)\right\|}$
γ_t	120	Weight for Diversity Guidance
β_1	1	Weight for OT Sampling Guidance
λ_1	1000	Entropy Regularization Weight
T	20	Sinkhorn Iterations, Same for Logit Matching
	Soft Label Relabeling	
Epochs	300, 500, 1000	300 for comparison with most baselines
Batch Size	50	Use 100 when IPC = 50
$\mathbb{T}(\text{IPC} = 10)$	ResNet-18, ShuffleNet	NA
$\mathbb{T}(\text{IPC} = 50)$	ResNet-18, MobileNet, EfficientNet, ShuffleNet	NA
	Student Model Logit Matching	
Optimizer	AdamW	NA
Learning Rate	0.001	Only use 1e-4 for Swin-Transformer
EMA Rate	0.99	Control EMA-based Evaluation
κ_1, κ_2	1, 0.025	Inherit from EDC
β_2	0.1	Weight for $\mathcal{L}_{\text{SD}}$
λ_2	0.1	Entropy Regularization Weight
Scheduler	Smoothing LR Schedule	$\zeta = 2$
Augmentation	RandomResizedCrop RandomHorizontalFlip	NA

G Further Experimental Analyses

G.1 More Sensitivity Analysis

In the main text, we presented a preliminary sensitivity analysis of key hyperparameters. To further evaluate their impact on performance, we conduct extensive ablation studies, with results summarized in Figures 4, 5, 6, 7 and Table 17. Overall, we observe that our method exhibits strong robustness to most hyperparameter settings: performance remains stable across a broad range of values. We select the hyperparameters by considering trade-offs among different architectures. Specifically, increasing the value of β_1 slightly improves the performance of ConvNet, but degrades that of ResNet-18. We therefore set $\beta_1 = 1$ to balance this trade-off. Similarly, increasing β_2 enhances performance on ResNet-AP, but negatively affects both ResNet-18 and ConvNet. Thus, we choose $\beta_2 = 0.1$ to

Table 14: Hyperparameter setting on ImageNette [49].

Config	Value	Explanation
	Guided Diffusion Sampling	
k	5	$\rho_t = k \cdot \sqrt{1-\alpha_t} \cdot \frac{\|\epsilon_\phi(\mathbf{z}_t, t, c)\|}{\|\nabla_{\mathbf{z}_t} \mathcal{G}_I(\hat{\mathbf{z}}_0 \mid t)\|}$
γ_t	50 when IPC=10 120 when IPC=50 or 100	Weight for Diversity Guidance
β_1	1	Weight for OT Sampling Guidance
λ_1	1000 when IPC=10 3000 when IPC=50 or 100	Entropy Regularization Weight
T	20	Sinkhorn Iterations, Same for Logit Matching
	Soft Label Relabeling	
Epochs	1000	Same for reproducing IGD
Batch Size	50 when IPC=10 100 when IPC=50 or 100	NA
$\mathbb{T}(\text{IPC} = 10)$	ResNet-18, MobileNet	NA
$\mathbb{T}(\text{IPC} = 100)$	ResNet-18, MobileNet, EfficientNet, ShuffleNet	Same for IPC=50
	Student Model Logit Matching	
Optimizer	AdamW	NA
Learning Rate	0.001	NA
EMA Rate	0.99	Control EMA-based Evaluation
κ_1, κ_2	1, 0.025	Inherit from EDC
β_2	0.1	Weight for $\mathcal{L}_{\text{SD}}$
λ_2	0.1	Entropy Regularization Weight
Scheduler	Smoothing LR Schedule	$\zeta = 2$
Augmentation	RandAugment RandomResizedCrop RandomHorizontalFlip	NA

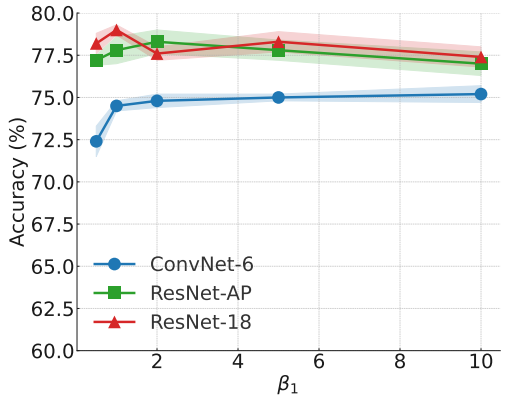


Figure 4: Effect of β_1 (OT sampling weight) on ImageNette [49] (IPC=10).

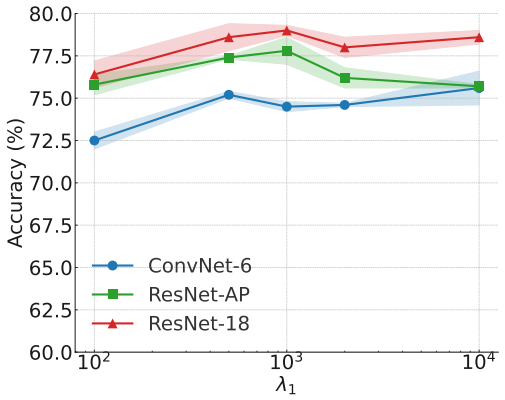


Figure 5: Effect of λ_1 (entropy regularization weight for sampling) on ImageNette [49] (IPC=10).

achieve the best average performance. In contrast, reducing either β_1 or β_2 consistently harms all model variants, which highlights the importance and effectiveness of our OT-based regularization terms. Since the latent and feature spaces differ in scale, we apply separate scaling factors λ_1 and λ_2 to normalize their contributions. As shown in the figures, setting $\lambda_1 = 1000$ and $\lambda_2 = 0.1$ yields a favorable trade-off across architectures. Taken together, our approach demonstrates two complementary aspects of robustness: (1) most OT-related hyperparameters exhibit consistent

Table 15: Hyperparameter setting on ImageNet-100 [13].

Config	Value	Explanation
	Guided Diffusion Sampling	
k	5	$\rho_t = k \cdot \sqrt{1-\alpha_t} \cdot \frac{\|\epsilon_\phi(\mathbf{z}_t, t, c)\|}{\|\nabla_{\mathbf{z}_t} \mathcal{G}_I(\hat{\mathbf{z}}_0 \| t)\|}$
γ_t	120	Weight for Diversity Guidance
β_1	1	Weight for OT Sampling Guidance
λ_1	1000	Entropy Regularization Weight
T	20	Sinkhorn Iterations, Same for Logit Matching
	Soft Label Relabeling	
Epochs	300	NA
Batch Size	100	NA
$\mathbb{T}(\text{IPC} = 10)$	ResNet-18, ShuffleNet	NA
$\mathbb{T}(\text{IPC} = 100)$	ResNet-18, MobileNet, EfficientNet, ShuffleNet	Same for IPC=50
	Student Model Logit Matching	
Optimizer	AdamW	NA
Learning Rate	0.001	NA
EMA Rate	0.99	Control EMA-based Evaluation
κ_1, κ_2	1, 0.025	Inherit from EDC
β_2	0.1	Weight for $\mathcal{L}_{\text{SD}}$
λ_2	0.1	Entropy Regularization Weight
Scheduler	Smoothing LR Schedule	$\zeta = 2$
Augmentation	RandomResizedCrop RandomHorizontalFlip	NA

Table 16: Hyperparameter setting on CIFAR-100 [21].

Config	Value	Explanation
	Guided Diffusion Sampling	
k	5	$\rho_t = k \cdot \sqrt{1-\alpha_t} \cdot \frac{\|\epsilon_\phi(\mathbf{z}_t, t, c)\|}{\|\nabla_{\mathbf{z}_t} \mathcal{G}_I(\hat{\mathbf{z}}_0 \| t)\|}$
γ_t	120	Weight for Diversity Guidance
β_1	1	Weight for OT Sampling Guidance
λ_1	1000	Entropy Regularization Weight
T	20	Sinkhorn Iterations, Same for Logit Matching
	Soft Label Relabeling	
Epochs	1000	NA
Batch Size	50	NA
$\mathbb{T}(\text{IPC} = 10)$	ResNet-18, ShuffleNet	NA
$\mathbb{T}(\text{IPC} = 100)$	ResNet18, ConvNet, MobileNet, WRN, ShuffleNet	Same for IPC=50
	Student Model Logit Matching	
Optimizer	AdamW	NA
Learning Rate	0.001	NA
EMA Rate	0.99	Control EMA-based Evaluation
κ_1, κ_2	1, 0.025	Inherit from EDC
β_2	0.1	Weight for $\mathcal{L}_{\text{SD}}$
λ_2	0.1	Entropy Regularization Weight
Scheduler	Smoothing LR Schedule	$\zeta = 2$
Augmentation	RandAugment RandomResizedCrop RandomHorizontalFlip	NA

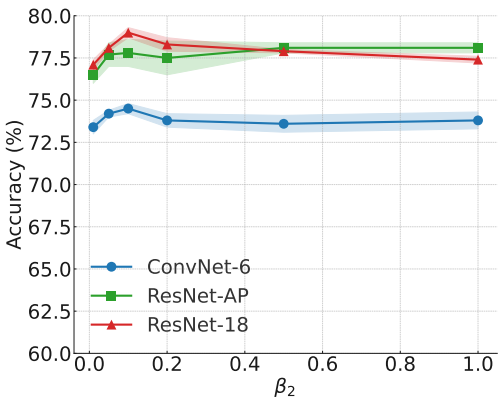


Figure 6: Effect of β_2 (OT matching weight) on ImageNette [49] (IPC=10).

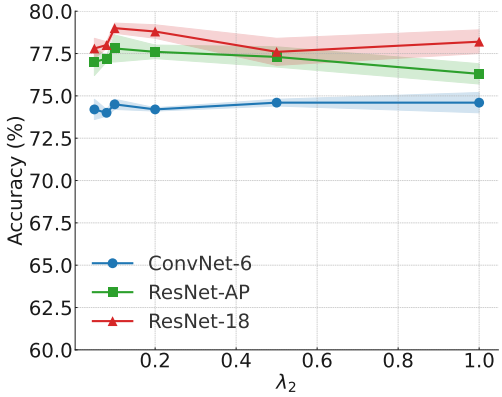


Figure 7: Effect of λ_2 (entropy regularization weight for logit matching) on ImageNette [49] (IPC=10).

Table 17: Effect of T on ImageNette [49] (IPC=10).

T	5	10	20	50	100
ResNet-18	77.9	78.6	79.0	78.8	78.9
ConvNet-6	73.8	74.3	74.5	74.7	74.6

behavior across different scenarios, requiring little to no manual adjustment, and in practice we only select λ_1 from the set $\{1000, 3000\}$ while keeping all other OT-related hyperparameters fixed (see Section F for default values); and (2) performance remains stable even when these parameters vary within reasonable ranges, eliminating the need of careful tuning.

G.2 Further Analysis of the Contraction Factor α

We provide a formal characterization and empirical analysis of the contraction factor α, which quantifies the degree to which soft labels reduce the discrepancy between the label and image distributions compared to hard labels. This factor plays a critical role in interpreting the effectiveness of soft supervision in dataset distillation. For efficient computation, we approximate the contraction factors using features in the latent space, enabling dimensionality reduction while preserving critical information.

Definition and Computation To compute α, we compare the class-conditional optimal transport (OT) distances from the real dataset to two variants of the distilled dataset: one annotated with soft labels $\nu_{\text{distill}}^{(\text{soft})}$ and one with hard labels $\nu_{\text{distill}}^{(\text{hard})}$. The contraction factor is then defined as the relative improvement in transport distance under soft supervision: $\alpha = \mathrm{W}(\mu_{\text{true}}, \nu_{\text{distill}}^{(\text{soft})}) / \mathrm{W}(\mu_{\text{true}}, \nu_{\text{distill}}^{(\text{hard})})$.

Let $\mathbf{Z}_{\mathcal{T}} \in \mathbb{R}^{N_1 \times d}$ and $\mathbf{Z}_{\mathcal{S}} \in \mathbb{R}^{N_2 \times d}$ denote latent embeddings extracted from real and distilled images, respectively. We construct the pairwise cost matrix using an ℓ_p norm:

$$\mathbf{C}_{ij} = \|\mathbf{Z}_{\mathcal{T}}(i) - \mathbf{Z}_{\mathcal{S}}(j)\|_p, \quad \mathbf{C} \in \mathbb{R}^{N_1 \times N_2}. \tag{19}$$

For each class $c \in \{1, \ldots, C\}$, we extract the marginal label distributions over samples: $\tilde{\mathbf{a}}^{(c)} = \mathbf{H}_{\mathcal{T}}[:, c]$ for real hard labels, and $\tilde{\mathbf{b}}^{(c)} = \mathbf{S}_{\mathcal{S}}[:, c]$ for distilled soft labels, where $\mathbf{H}_{\mathcal{T}} \in \{0, 1\}^{N_1 \times C}$ and $\mathbf{S}_{\mathcal{S}} \in [0, 1]^{N_2 \times C}$. These are normalized into valid probability vectors:

$$\mathbf{a}^{(c)} = \frac{\tilde{\mathbf{a}}^{(c)}}{\sum_i \tilde{a}_i^{(c)}}, \quad \mathbf{b}^{(c)} = \frac{\tilde{\mathbf{b}}^{(c)}}{\sum_j \tilde{b}_j^{(c)}}. \tag{20}$$

We then perform entropic regularized OT using the Sinkhorn algorithm. The Gibbs kernel is defined as:

$$\mathbf{K} = \exp\left(-\frac{\mathbf{C}}{\varepsilon}\right), \tag{21}$$

Table 18: Effect of α on ImageNet-1K [22] (IPC=10). **Config A:** ResNet-18. **Config B:** ResNet-18, MobileNet, EfficientNet, ShuffleNet. **Config C:** ResNet-18, MobileNet, AlexNet, ShuffleNet. **Config D:** ResNet-18, ShuffleNet.

Teachers	Config A	Config B	Config C	Config D
α	1.00	0.99	0.95	0.93
Acc (ResNet-18)	50.3	52.3	52.7	52.9
Acc (Swin)	47.2	47.8	49.2	50.2

Table 19: Effect of α on ImageNet-1K [22] (IPC=50). **Config A:** ResNet-18. **Config B:** ResNet-18, MobileNet, EfficientNet, ShuffleNet. **Config C:** ResNet-18, MobileNet, AlexNet, ShuffleNet. **Config D:** ResNet-18, ShuffleNet.

Teachers	Config A	Config B	Config C	Config D
α	0.97	0.16	0.97	1.00
Acc (ResNet-18)	62.3	61.9	60.8	60.5
Acc (Swin)	65.5	68.2	65.5	65.3

where ε controls regularization strength. The scaling vectors $\mathbf{u}$ and $\mathbf{v}$ are initialized uniformly as:

$$\mathbf{u}^0 \leftarrow \mathbf{1}/N_1, \quad \mathbf{v}^0 \leftarrow \mathbf{1}/N_2, \tag{22}$$

and iteratively updated as:

$$\mathbf{u}^{t+1} = \frac{\mathbf{a}^{(c)}}{\mathbf{K}\mathbf{v}^t + \delta}, \tag{23}$$

$$\mathbf{v}^{t+1} = \frac{\mathbf{b}^{(c)}}{\mathbf{K}^\top \mathbf{u}^{t+1} + \delta}, \tag{24}$$

where δ ensures numerical stability. After T iterations, the transport plan is:

$$\gamma^{(c)} = \mathrm{diag}(\mathbf{u}) \cdot \mathbf{K} \cdot \mathrm{diag}(\mathbf{v}), \tag{25}$$

and the classwise OT cost becomes:

$$\mathcal{L}_c = \langle \gamma^{(c)}, \mathbf{C} \rangle = \sum_{i,j} \gamma_{ij}^{(c)} C_{ij}. \tag{26}$$

Averaging over the valid class set $\mathcal{C}$ (i.e., classes with non-zero support in both distributions) yields:

$$\mathrm{W}(\mu_{\text{true}}, \nu_{\text{distill}}^{(\text{soft})}) = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \mathcal{L}_c. \tag{27}$$

To compute the counterpart $\mathrm{W}(\mu_{\text{true}}, \nu_{\text{distill}}^{(\text{hard})})$, we replace $\mathbf{S}_\mathcal{S}$ with its hard label projection $\mathbf{S}_\mathcal{S}^h \in \{0,1\}^{N_2 \times C}$ and repeat the same computation.

Empirical Insights We conduct several additional experiments, with results shown in Tables 18 and 19. Our empirical analysis provides several important observations regarding the role of the contraction factor α in guiding effective distillation. We first find that α is highly sensitive to the diversity and calibration quality of the teacher ensemble, and this sensitivity is modulated by the IPC (images per class) setting. In low-IPC regimes (e.g., IPC=10), using overly complex or inconsistent teacher predictions increases the optimal transport distance $\mathrm{W}(\mu_{\text{true}}, \nu_{\text{distill}}^{(\text{soft})})$, leading to smaller α values and ultimately harming the generalization ability of the distilled dataset. Conversely, when IPC is sufficiently high (e.g., IPC=50), stronger and more expressive teacher distributions better capture the semantic structure of the real data, resulting in larger α values and improved alignment between real and synthetic distributions.

Second, we observe that deliberately reducing α, thereby explicitly minimizing the overall optimal transport distance, leads to significant improvements in downstream model performance. This effect is particularly evident under both settings, where configurations with smaller α (e.g., Config D)

Table 20: Comparison of our optimal transport-based distance with other measures for guided diffusion sampling on ImageNet-1K [22].

IPC	MMD	MMD (RKHS)	Ours
10	49.4	50.3	52.9
50	60.4	60.6	61.9

Table 21: Comparison between the sample-wise and batch-wise optimal transport distance for student model logit matching on ImageNet-1K [22]. OOM: CUDA out of memory.

Level	ResNet-18		MobileNet-V2	
	IPC=10	IPC=50	IPC=10	IPC=50
Sample-wise	50.6	60.8	OOM	OOM
Batch-wise	52.9	61.9	51.0	61.0

achieve better Top-1 accuracy across both the ResNet-18 and Swin Transformer. These results empirically confirm that shrinking the distributional gap through controlling α facilitates more efficient and effective knowledge transfer. Third, in high-IPC regimes, when a single teacher is used (e.g., Config A), student models that share the same architecture as the teacher can fully exploit the teacher's architectural biases, achieving strong performance. However, such tight alignment may limit generalization to unseen architectures. By appropriately contracting α, we encourage the distilled dataset to encode more transferable, architecture-independent features, thereby improving the student's adaptability to diverse downstream architectures. Overall, these findings validate α as a principled and tunable indicator of distillation quality, and highlight the importance of strategic contraction strategies tailored to both teacher complexity and downstream generalization targets.

G.3 Comparison with Other Distance Measure

For guided diffusion sampling Unlike conventional metrics such as cosine similarity, KL divergence, or mean squared error, which rely on explicit instance-level alignment, optimal transport (OT) enables distribution-level alignment without enforcing one-to-one correspondences. This property makes OT particularly well-suited for dataset distillation scenarios, where the number of synthetic samples is significantly smaller than that of the original dataset, and direct pairing is often infeasible or suboptimal. While Maximum Mean Discrepancy (MMD)-based measures have also been adopted for distribution alignment without requiring exact correspondences, they primarily focus on matching global distributional statistics and fail to capture fine-grained pairwise relations between individual instances in the real and synthetic distributions. In contrast, our OT-based formulation explicitly models such pairwise interactions and thus facilitates more accurate and semantically consistent guidance during the diffusion sampling process. As shown in Table 20, our method consistently outperforms MMD and MMD with reproducing kernel Hilbert spaces (RKHS) baselines on ImageNet-1K under both low-IPC and high-IPC settings. These results underscore the importance of modeling instance-level correspondences for effective guidance and highlight the superiority of OT in capturing the geometry of complex data distributions.

For student model logit matching Table 21 illustrates the consistent superiority of batch-wise OT distance over sample-wise OT distance. This result highlights that batch-wise Sinkhorn distance is more effective in transferring the distributional geometry captured by the distilled set from label-image space to the newly trained student models. The sample-wise logit matching approach treats each instance independently, failing to account for the global structure and correlations within a batch.

Table 22: Performance comparison of logit matching methods on ImageNette (IPC=10).

Network	ConvNet-6	ResNetAP-10	ResNet-18
MMD	72.6	75.3	76.4
KL	73.0	75.4	77.6
OTM	**74.5**	**77.8**	**79.0**

Table 23: Distribution coverage comparison among different methods.

Threshold	DiT [44]	DiT-IGD [24]	Ours
10	40.2	40.8	41.6
12	54.6	56.3	57.5

In contrast, our batch-wise formulation preserves inter-sample relationships, enabling more faithful distributional alignment and resulting in more robust knowledge transfer. Moreover, when dealing with datasets containing a large number of classes (e.g., 1,000 classes in ImageNet-1K), the batch-wise approach substantially reduces memory consumption and avoids the CUDA out-of-memory issues frequently encountered by sample-wise matching, further enhancing its scalability.Also, although KL and MMD serve as simpler divergences, they are inherently limited. KL divergence is applied per sample and ignores inter-sample relationships, while MMD matches only global statistics. In contrast, our OTM applies batch-wise OT alignment between student logits and soft labels, capturing the joint distributional structure of samples. This enables OT to faithfully preserve inter-sample geometry and match structural uncertainty in the soft labels, which KL and MMD overlook. As shown in Table 22, this leads to a clear performance gain:

G.4 More Discussions on Runtime

While EDC [23] reduces the runtime during the recovery phase compared to previous work, it does not fully optimize for multi-GPU parallelism across its various processes. Specifically, the pre-sampling and post-sampling phases during initialization do not benefit from multi-GPU parallelism, as parallelizing these steps does not result in substantial time reduction. Moreover, the recovery phase is inherently constrained by data loading and model-inversion methods, and beyond four GPUs, further increases in parallelism yield minimal improvements in runtime. In contrast, our approach is designed to optimize each class separately, with the sampling process dependent only on images sampled from the same class and the corresponding real images. As a result, our method scales more efficiently with the number of GPUs, with runtime decreasing nearly inversely proportional to the number of GPUs. Furthermore, when the need for high IPC arises, our method can be adapted to split high-IPC tasks into several lower-IPC ones for parallel processing, maintaining strong parallel efficiency and further enhancing its applicability in real-world scenarios.

G.5 Data Coverage Analysis

To assess the representational fidelity of the distilled dataset, we adopt a coverage-based evaluation metric. Specifically, for each data point in the original dataset, we determine whether it has at least one nearest neighbor in the distance dataset within a predefined distance threshold. This metric reflects how well the surrogate data captures the underlying structure of the original distribution. As shown in Table 23, our method consistently achieves higher coverage compared to baseline methods across multiple thresholds. The improvements are observed over both the original DiT [44] model and DiT-IGD [24], indicating that our approach provides better distributional alignment. Notably, the performance gap widens as the threshold increases, further validating the robustness of our distilled data in covering diverse modes of the original dataset.

G.6 Comparison with DWA

DWA [89] enhances diversity by adjusting the statistics of the squeezed network based on each generated sample. However, it still relies solely on global statistics, specifically the mean and variance associated with batch normalization (BN), and thus fails to capture the rich instance-level information and geometric distributional structures inherent in the real dataset. Visualizations from the DWA paper further illustrate that, while the directed weight adjustment improves the diversity of the distilled dataset, the distribution remains concentrated, failing to adequately cover the majority of the real data distribution. In Table 24, we compare our method with DWA across multiple student models, and the results clearly demonstrate a significant performance advantage of our approach. This further emphasizes the importance of leveraging fine-grained instance-level information for achieving improved model performance and more faithful distributional alignment.

Table 24: Comparison with DWA [89] on ImageNet-1K [22].

Method	ResNet-18		MobileNet-V2		EfficientNet-B0	
	IPC10	IPC50	IPC10	IPC50	IPC10	IPC50
DWA [89]	37.9±0.2	55.2±0.2	29.1±0.3	51.6±0.5	37.4±0.5	56.3±0.4
Ours	**52.9±0.1**	**61.9±0.5**	**51.0±0.6**	**61.0±0.4**	**56.7±0.2**	**64.4±0.1**

Table 25: Comparison between WMDD [43] and our method on ImageNette [49] and ImageNet-1K [22] under different IPC settings.

Dataset	ImageNette		ImageNet-1K	
IPC	10	50	10	50
WMDD	64.8±0.4	83.5±0.3	38.2±0.2	57.6±0.5
Ours	**79.0±0.3**	**89.3±0.3**	**52.9±0.1**	**61.9±0.5**

G.7 Comparison with WMDD

Although both our method and WMDD [43] utilize optimal transport (OT), they differ significantly in both methodology and motivation, leading to distinct formulations and implementations.

WMDD [43] is a distribution-matching-based distillation method that applies OT in a single, offline step to compute a Wasserstein barycenter over the real data's feature distribution. This barycenter is then used as a fixed target throughout training, where synthetic images are optimized to match it using a standard L2 loss in the feature space. In contrast, we introduce a fundamentally different generative paradigm where OT is not a static, one-off computation, but a dynamic guidance mechanism integrated throughout the entire data synthesis and training pipeline. Specifically, OT guides the sampling of synthetic images by aligning latent representations, regulates the soft label relabeling process by matching label complexity to the image distribution, and structures the training loss of the student model by aligning its logits to the relabeled targets.

The motivation behind WMDD is to replace the use of simple data summaries, such as the feature means often targeted by MMD-based methods, with a more geometrically meaningful summary, namely the Wasserstein barycenter, derived from the Wasserstein metric. In contrast, our method is driven by the need to address inherent limitations in generative distillation pipelines, which often fail to preserve the fine-grained geometry of the real data distribution—particularly intra-class variations and local modes. These aspects are explicitly addressed in our framework through a multi-stage, OT-guided design.

We compare the top-1 accuracy of our method with WMDD under different images-per-class (IPC) settings on both the ImageNette and ImageNet-1K datasets in Table 25.

G.8 Robustness Evaluation

To assess the robustness of student models trained with distilled datasets, we follow the evaluation protocol established in DD-RobustBench [90], utilizing adversarial attacks implemented in the TorchAttacks library [91]. As shown in Tables 26 and 27, we evaluate models trained on ImageNette [49] under IPC=10 and IPC=50 settings, measuring both standard test accuracy and adversarial robustness against a variety of attack methods.

Our method consistently achieves higher clean accuracy and substantially improves robustness compared to MTT [14] across different perturbation budgets ($|\varepsilon| = 4/255$ and $|\varepsilon| = 8/255$). These improvements can be attributed to the distributional properties enforced by our optimal transport (OT)-based distillation framework. By minimizing the OT distance between the synthetic and real data distributions, our method preserves not only class-level statistics but also fine-grained, instance-level geometric structures. This leads to the learning of semantically faithful and smoother decision boundaries, which are inherently more resilient to adversarial perturbations. Moreover, our OT-guided diffusion sampling produces visually more coherent and perceptually realistic images compared to other types of approaches. The generated synthetic samples better preserve the semantic integrity and natural variability of the original data, providing stronger perceptual signals during model training.

As a result, the student model benefits from a more robust feature space that aligns well with human perception, further enhancing adversarial robustness beyond purely decision-boundary-level effects.

In contrast, methods that primarily match global statistics or rely on heuristic trajectory guidance, such as MTT, often produce synthetic datasets lacking such structural fidelity, resulting in brittle decision boundaries that are more vulnerable to attacks.

From a theoretical perspective, prior works [92, 93] have established a strong connection between adversarial robustness and the sharpness of decision boundaries: sharper, more irregular boundaries tend to amplify adversarial vulnerability, whereas flatter, smoother boundaries promote robustness. By aligning not only global distributions but also the local transportation cost between real and synthetic samples, OT encourages the distilled student model to form flatter and more coherent decision surfaces aligned with the real data geometry.

Moreover, from a loss landscape perspective, minimizing the OT distance guides optimization towards flatter minima, where small input perturbations induce minimal output changes. This connection is well supported by prior studies [94, 95], which show that flatter loss surfaces correlate strongly with improved adversarial robustness. Together, these empirical and theoretical insights demonstrate that preserving distributional geometry via optimal transport provides a principled and effective pathway for enhancing the adversarial robustness of models trained on distilled datasets.

Table 26: Performance comparison on DD-RobustBench [90] evaluated on ImageNette [49], under a perturbation budget of $|\varepsilon| = 4/255$. Results for MTT [14] are directly copied from the DD-RobustBench benchmark.

Attack Methods	IPC=10		IPC=50	
	MTT [14]	Ours	MTT [14]	Ours
Clean Accuracy	66.4	**69.1**	67.7	**84.6**
FGSM	10.8	**20.8**	8.4	**24.0**
PGD	4.6	**9.2**	2.6	**9.8**
CW	4.6	**12.0**	1.4	**14.8**
VMI	5.4	**9.0**	2.0	**11.2**
Jitter	12.2	**20.4**	13.0	**23.8**

Table 27: Performance comparison on DD-RobustBench [90] evaluated on ImageNette [49], under a perturbation budget of $|\varepsilon| = 8/255$. Results for MTT [14] are directly copied from the DD-RobustBench benchmark.

Attack Methods	IPC=10		IPC=50	
	MTT [14]	Ours	MTT [14]	Ours
Clean Accuracy	66.4	**69.1**	67.6	**84.6**
FGSM	0.8	**11.0**	1.8	**14.8**
PGD	0.2	**2.8**	1.2	**15.0**
CW	0.2	**9.6**	0.2	**6.8**
VMI	0.2	**0.8**	0.2	**2.0**
Jitter	11.4	**12.4**	9.8	**14.6**

G.9 Evaluation on Low-IPC Settings

We have conducted additional experiments on ImageNet-1K [22] for the challenging settings of IPC=1, IPC=2, and IPC=5. The results of these new experiments are presented in the Table 28.

Importantly, in the IPC=1 setting, since only one synthetic image is generated per class, the OTG process cannot leverage previously distilled samples for alignment. Instead, for each class, we compute the OT distance between its single synthetic candidate and the corresponding real images in the latent space to guide generation.

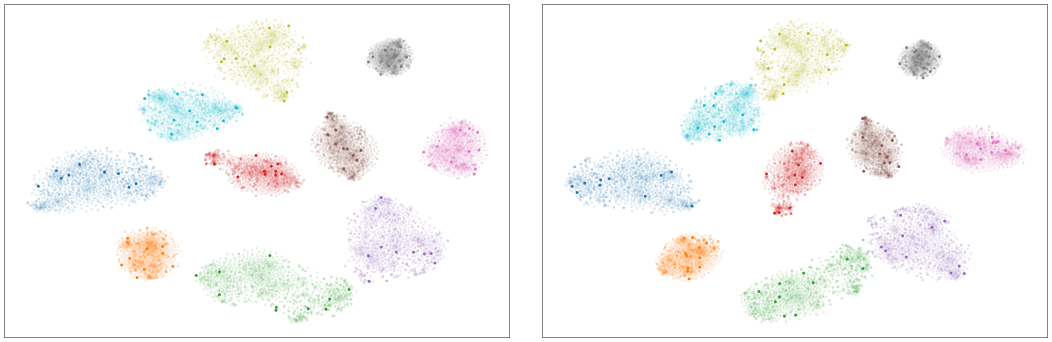
Table 28: Performance comparison of different methods on ImageNet-1K under small IPCs (1, 2, 5). Best results are in bold.

IPC	Method							
	DM	FrePo	TESLA	SRe2L	RDED	EDC	DiT-IGD	Ours
1	1.5	7.5	7.7	0.4	6.6	12.8	10.7	**15.9**
2	1.7	9.7	10.5	–	16.5	22.8	20.6	**25.9**
5	–	–	–	–	23.8	39.5	38.6	**45.7**

H More Visualization Results

H.1 T-SNE Results

To assess the effectiveness of our OT-guided diffusion sampling, we present the t-SNE [96] results in Figure 8. The diversity in IGD is driven solely by cosine-similarity based diversity guidance, without leveraging the distributional structure of the real dataset. This limitation leads to insufficient coverage of critical regions in the true data distribution, such as the central region of the green (Cassette player), the lower part of the blue (Tench), and the middle-upper section of the purple (Church) areas. Consequently, several important subclasses are absent from the distilled dataset, resulting in the new model failing to learn relevant intra-class variations and important subclass-specific information. In contrast, our approach iteratively computes the optimal transport distance between the real dataset and the distilled set, explicitly incorporating both intra-class structures and finer substructures of the real data. This enables our distilled dataset to capture a broader range of essential submodalities and regions, facilitating a more comprehensive transfer of information to the new model, and minimizing information loss. By employing the optimal transport distance as an additional supervision signal during the new model's training, we ensure the effective transfer of this enriched information, leading to significant improvements in model performance.



(a) IGD (b) Ours

Figure 8: Visualization study for sample distributions of distilled datasets (IPC=10) generated by IGD [24] and Ours versus the original ImageNette [49] dataset. The dark points represent the distilled set, while the light points represent the real (original) set. The diversity in IGD [24] is driven solely by random diversity guidance, lacking awareness of the real data distribution. As a result, it fails to cover critical regions such as such as the central region of the green class (Cassette player), the lower part of the blue class (Tench), and the middle-upper section of the purple class (Church). In contrast, our method incorporates both intra-class structures and fine-grained substructures of the real data, which allows it to effectively cover most subclass regions.

H.2 Distilled Images

Figures 9 and 10 provide additional visual comparisons between IGD [24] and our method, as well as standalone visualizations of our distilled dataset. Our method effectively captures the structural information of the real data distribution, resulting in high-fidelity samples with semantic diversity that faithfully reflects the underlying real-world distribution.

Figure 9: Additional visualization comparing the distilled datasets generated by IGD [24] and our approach on ImageNet-1K [22] (IPC=10).

Figure 10: Additional visualizations for our distilled set on ImageNet-1K [22] (IPC=10).

I Limitations

Our current framework inherits the trajectory influence guidance mechanism from IGD [24], which, while effective for improving the general and global alignment of sampled data, introduces substantial computational overhead. Specifically, the additional sampling steps required to maintain trajectory consistency significantly slow down the generation process compared to the vanilla DiT [44], which operates without such constraints. In future work, we aim to reformulate the guidance process to retain benefits while reducing the reliance on explicit trajectory tracking, thereby enabling faster and more scalable sampling.

J Broader Impact

Our work aims to reduce dataset size while maintaining performance, enabling model training with significantly lower computational and storage costs. This can lower the entry barrier for institutions with limited resources and promote environmentally sustainable AI development [97]. Moreover, our distilled datasets have the potential to facilitate efficient learning in federated and continual learning scenarios, thereby enhancing data privacy and supporting model adaptation across distributed systems. However, as with most data-driven approaches, there exists a risk that the distilled data may retain or amplify biases present in the original datasets. This could lead to unintended consequences, particularly in sensitive applications. Additionally, by accelerating the deployment of compact models, our method may inadvertently contribute to insufficiently audited systems being widely adopted. We emphasize the importance of responsible deployment, including bias auditing, fairness-aware design, and transparency, and encourage future work to explore these aspects more thoroughly.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Our abstract and introduction accurately reflect the paper's contributions to generative large-scale dataset distillation.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Yes, we discuss this in Appendix I.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

Justification: This paper does not include theorems or lemmas.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer:[Yes]

   Justification: Yes, we provide sufficient implementation details in in Section 5.1 and Appendix F.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We provide code in the supplementary materials, and provide anonymous Github link.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Experimental setting is described in Section 5.1 and Appendix F.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We report error bars in Tables 1, 2, 3, 4, 5, 6.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide the computer resouces in Section 5.1.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Our research conform with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss the broader impact in Appendix J.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All assets used in the paper, including datasets, are publicly available. Proper credits are given to the creators or original owners of these datasets where applicable. The licenses and terms of use for these datasets are explicitly mentioned and respected in accordance with their respective guidelines.

    Guidelines:

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

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We have attached our code and user instructions in the supplementary materials

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

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