Information-Guided Diffusion Sampling for Dataset Distillation

Linfeng Ye
University of Toronto
linfeng.ye@mail.utoronto.ca

Shayan Mohajer Hamidi
Stanford University
smohajer@stanford.edu

Guang Li[*]
Hokkaido University
guang@lmd.ist.hokudai.ac.jp

Takahiro Ogawa
Hokkaido University
ogawa@lmd.ist.hokudai.ac.jp

Miki Haseyama
Hokkaido University
mhaseyama@lmd.ist.hokudai.ac.jp

Konstantinos N. Plataniotis
University of Toronto
kostas@comm.utoronto.ca

Abstract

Dataset distillation aims to create a compact dataset that retains essential information while maintaining model performance. Diffusion models (DMs) have shown promise for this task but struggle in low images-per-class (IPC) settings, where generated samples lack diversity. In this paper, we address this issue from an information-theoretic perspective by identifying two key types of information that a distilled dataset must preserve: (i) *prototype information* $\mathrm{I}(X;Y)$, which captures label-relevant features; and (ii) *contextual information* $\mathrm{H}(X|Y)$, which preserves intra-class variability. Here, (X, Y) represents the pair of random variables corresponding to the input data and its ground truth label, respectively. Observing that the required contextual information scales with IPC, we propose maximizing $\mathrm{I}(X;Y) + \beta\mathrm{H}(X|Y)$ during the DM sampling process, where β is IPC-dependent. Since directly computing $\mathrm{I}(X;Y)$ and $\mathrm{H}(X|Y)$ is intractable, we develop *variational estimations* to tightly lower-bound these quantities via a data-driven approach. Our approach, information-guided diffusion sampling (IGDS), seamlessly integrates with diffusion models and improves dataset distillation across all IPC settings. Experiments on Tiny ImageNet and ImageNet subsets show that IGDS significantly outperforms existing methods, particularly in low-IPC regimes.

1 Introduction

The success of high-performance deep neural networks (DNNs) is largely attributed to large-scale, highly informative datasets [LeCun et al., 2015]. However, the size of these datasets poses a substantial burden on storage and computational resources during model training [Deng et al., 2009, Salamah et al., 2024, Ye et al., 2022, Kaplan et al., 2020]. To mitigate the cost of training DNNs, dataset distillation [Wang et al., 2018, Sachdeva and McAuley, 2023] has been extensively studied in recent years as a potential solution to compress datasets, thereby reducing both storage requirements and computational costs. In this approach, a smaller dataset whose compactness is typically measured by images-per-class (IPC) is constructed as a substitute for the original dataset, while still enabling the trained model to achieve decent generalization performance on unseen test data points.

[*] Correspondence to: Guang Li <guang@lmd.ist.hokudai.ac.jp>

Accepted at third SPIGM workshop @ NeurIPS 2025

To construct such a compact dataset, the distillation process typically involves an iterative optimization of pixel values and auxiliary model weights to align either with the model's weight trajectory [Zhao et al., 2021, Hamidi and Ye, 2025, Cazenavette et al., 2022] or feature statistics [Wang et al., 2022, Deng et al., 2024, Sajedi et al., 2023]. However, this approach has two major drawbacks: (i) High computational cost—most existing methods require jointly optimizing auxiliary model parameters and distilled samples at the pixel level through an iterative process, resulting in significant computational overhead. (ii) Poor generalization across different model architectures—the performance of models trained on the distilled dataset is highly dependent on the architecture of the auxiliary DNN. When the target model's architecture differs from that of the auxiliary DNN, significant performance degradation is often observed.

To overcome these drawbacks, recent studies have proposed generative distillation [Zhao and Bilen, 2022, Zhong et al., 2022, Cazenavette et al., 2023], which leverages a generative model to synthesize a new, compact dataset. In this approach, a generative model is trained on the target dataset and then used to sample distilled data [Zhao and Bilen, 2022, Chi et al., 2024, Li et al., 2024]. Consequently, the resulting distilled dataset is both model-architecture-agnostic and more efficiently generated. Among generative models, diffusion models (DMs) have emerged as a strong choice for dataset distillation, demonstrating state-of-the-art performance in this setting [Gu et al., 2024a, Su et al., 2024, Hamidi and Ye, 2024, Chi et al., 2025a, Chen et al., 2025, Li et al., 2025]. Nonetheless, DM-based dataset distillation suffers from poor performance in low-IPC scenarios, where the number of IPCs is small. In these cases, the accuracy is nearly as low as training on a randomly chosen subset. A primary reason is that, under low-IPC conditions, the model's generated samples reflect only part of the true data distribution, leading to a distilled dataset with limited diversity and substantial information loss. This shortfall grows more severe as the IPC decreases.

To address this limitation and enable the generation of informative samples, we first seek to quantify the essential information that must be preserved. To this end, we adopt an information-theoretic perspective [Shannon, 1948, Wu et al., 2024, Yang and Ye, 2024, Cover and Thomas, 2006, Yang et al., 2024]. Specifically, we quantify the relevant information using the Shannon entropy $\mathrm{H}(\cdot)$ [Shannon, 1948] on the random variable (RV) X, which represents the input dataset. We then expand $\mathrm{H}(X)$ as: $\mathrm{H}(X) = \mathrm{I}(X;Y) + \mathrm{H}(X|Y)$, where Y is an RV denoting the ground-truth (GT) label[2]. Through this decomposition, the total information in $\mathrm{H}(X)$ naturally splits into: (i) *prototype information* $\mathrm{I}(X;Y)$, reflecting how much information X provides about its GT label; and (ii) *contextual information* $\mathrm{H}(X|Y)$, capturing the remaining information in X once its GT label is given.

A successful dataset distillation should preserve both the prototype and contextual information of the target dataset. However, we observe that the required amount of contextual information depends on the IPC: a higher (resp. lower) IPC necessitates more (resp. less) contextual information. Building on this insight, we propose to maximize $\mathrm{I}(X;Y) + \beta\mathrm{H}(X|Y)$ during the DM sampling process, where β is selected according to the IPC. Since directly computing $\mathrm{I}(X;Y)$ and $\mathrm{H}(X|Y)$ is intractable, we develop *variational estimations* to tightly lower-bound these quantities via a data-driven approach. Specifically, we train a DNN using a novel training algorithm that provides tight lower bounds on both $\mathrm{I}(X;Y)$ and $\mathrm{H}(X|Y)$. We refer to this DNN as a variational estimator (VE). Once the VE is trained, it is frozen and used in the DM sampling process, guiding the generation of distilled data that maximally preserves both prototype and contextual information.

Our work introduces a novel dataset distillation approach, Information-Guided Diffusion Sampling (IGDS), which leverages information-theoretic principles to enhance the effectiveness of diffusion models in low IPC settings. The key contributions of this paper are as follows:

• We introduce a principled framework based on Shannon entropy decomposition, identifying prototype information $\mathrm{I}(X;Y)$ and contextual information $\mathrm{H}(X|Y)$ as crucial components for effective dataset distillation. Our approach dynamically balances these terms to optimize the informativeness of the distilled dataset.

• Since directly computing prototype information $\mathrm{I}(X;Y)$ and contextual information $\mathrm{H}(X|Y)$ is intractable, we develop a data-driven VE using deep neural networks to obtain tight lower bounds on these quantities. This estimator is seamlessly integrated into the diffusion sampling process.

[2] We use "ground truth" and "prototype" interchangeably in this paper.

$$\begin{tikzcd}
Y \arrow[r, "\text{Sampling according to } P_{X|Y}(\cdot\,|y)"'] & X \arrow[r, "\text{Encoder } f_{\boldsymbol{\theta}}(\cdot)"'] & \hat{X} \arrow[r, "\text{Classifier } g_{\boldsymbol{\psi}}(\cdot)"'] & \hat{Y}
\end{tikzcd}$$

Figure 1: Multi-class classification can be modeled as a Markov chain. Sample X is sampled from the class Y, according to the $P_{X|Y}(\cdot|y)$. The encoder then maps the X to the feature representation $\hat{X}$, which is further mapped by the classifier to an output probability vector $\hat{Y}$.

• We propose IGDS, a novel diffusion-based dataset distillation method that maximizes $\mathrm{I}(X;Y) + \beta\mathrm{H}(X|Y)$ during the sampling process. The weight β is IPC-dependent, allowing for adaptive control over prototype and contextual information retention.

• Extensive experiments on Tiny ImageNet [Le and Yang, 2015] and subsets of ImageNet [Deng et al., 2009] demonstrate that IGDS achieves superior performance compared to existing methods, particularly in low-IPC scenarios, where prior diffusion-based approaches suffer from poor diversity and high information loss.

2 Notation

For a positive integer C, let $[C] \triangleq \{1, \dots, C\}$. Denote by $P[i]$ the i-th element of the vector P. For two vectors U and V, denote by $\langle U, V \rangle$ their inner product. For two matrices $M \in \mathbb{R}^{m \times n}$ and $N \in \mathbb{R}^{n \times k}$, denote by $M@N$ their matrix product. We use $|\mathcal{C}|$ to denote the cardinality of a set $\mathcal{C}$. The entropy of C-dimensional probability vector P is defined as $\mathrm{H}(P) = \sum_{c=1}^{C} -P[c] \log P[c]$. Also, the cross entropy of two C-dimensional probability vectors P_1 and P_2 is defined as $\mathrm{CE}(P_1, P_2) = \sum_{c=1}^{C} -P_1[c] \log P_2[c]$, and their Kullback–Leibler (KL) divergence is defined as $\mathrm{KL}(P_1 || P_2) = \sum_{c=1}^{C} P_1[c] \log \frac{P_1[c]}{P_2[c]}$. For a random variable X, denote by P_X its probability distribution, and by $E_X[\cdot]$ the expected value w.r.t. X. The mutual information between two random variables X and Y is written as $\mathrm{I}(X;Y)$, and the conditional mutual information of X and Y given a third random variable Z is $\mathrm{I}(X;Y|Z)$. Consider a dataset $\mathcal{D}$ of size n with C classes, $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, where each $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in [C]$. For any class y, we define $\mathcal{D}_y = \{(\boldsymbol{x}_j, y_j) \in \mathcal{D} | y_j = y\}$ the subset of $\mathcal{D}$ containing all samples with label y. Lastly, the softmax operation is denoted by $\sigma(\cdot)$.

3 Methodology

As discussed in section 1, the performance of DM-based dataset distillation degrades significantly when the IPC is small. In such low-IPC conditions, the DM tends to produce samples that represent only a portion of the true data distribution, omitting many essential modes. As a result, the distilled dataset exhibits limited diversity and loses substantial information about the underlying classes. Empirically, this often manifests in accuracies that are comparable to training on a mere random subset of the original dataset. The challenge intensifies with decreasing IPC, since fewer examples per class mean the generative process has even less guidance for reproducing the full range of relevant features. In practice, these shortcomings severely limit the applicability of DM-based distillation for real-world tasks where one cannot afford to collect a large number of samples for each class.

To tackle this limitation, it is crucial to identify the core information that must be retained in the distilled dataset. To this end, we adopt an information-theoretic perspective to rigorously quantify and preserve this information. Specifically, we measure the total information in the input dataset, represented as a random variable X, using the Shannon entropy $\mathrm{H}(X)$. Noting that a significant part of the value of a dataset lies in its ability to determine labels, we decompose $\mathrm{H}(X)$ as follows

$$\mathrm{H}(X) = \underbrace{\mathrm{I}(X;Y)}_{\text{prototype information}} + \underbrace{\mathrm{H}(X|Y)}_{\text{contextual information}}, \tag{1}$$

where Y is a random variable denoting the GT label. This decomposition distinctly separates prototype information $\mathrm{I}(X;Y)$, which quantifies how much X reveals about its label, from contextual information $\mathrm{H}(X|Y)$, which captures the variability and richness of the data given its label [Yang et al., Chi et al., 2025b]. In other words, prototype information ensures that the distilled dataset remains discriminative for classification tasks, whereas contextual information guards against the collapse into a narrow subset of features, thus maintaining diversity and nuance (in section 4.2, we visualize the semantic meaning of prototype and contextual information). By explicitly accounting for both these components, we can better capture the data's essential characteristics, even in low-IPC regimes. A successful dataset distillation scheme must retain both prototype information, ensuring that each class is accurately characterized, and contextual information, preserving the variety and richness of the underlying data distribution. However, our observations indicate that the requisite amount of contextual information scales with the IPC: higher IPC scenarios allow, and indeed necessitate, more contextual detail, whereas lower IPC settings benefit more from a tighter focus on prototype information (please see section 3.3 for additional details).

Guided by this insight, we aim to balance these two information types by maximizing the objective

$$\mathrm{I}(X;Y) \; + \; \beta\,\mathrm{H}(X|Y), \tag{2}$$

where the scalar $\beta > 0$ is chosen to reflect the IPC: a larger β for high-IPC settings increases the emphasis on contextual richness, while a smaller β in low-IPC scenarios prioritizes critical prototype information.

Computing $\mathrm{I}(X;Y)$ and $\mathrm{H}(X|Y)$ is challenging, and to the best of our knowledge, no previous work has accomplished this. To overcome this difficulty, we introduce a novel method in section 3.1 that provides variational estimates for these quantities. Subsequently, in section 3.2, we leverage these estimates to guide the sampling process of diffusion models.

The proofs for all propositions and theorems presented in this paper are deferred to section B.

3.1 Variational Estimates for $\mathrm{I}(X;Y)$ and $\mathrm{H}(X|Y)$

Since directly estimating the $\mathrm{H}(X|Y)$ and $\mathrm{I}(X;Y)$ is infeasible, we employ an auxiliary DNN composed of an encoder $f_{\boldsymbol{\theta}}(\cdot)$ and a classifier $g_{\boldsymbol{\psi}}(\cdot)$ to help us finding variational estimates for both $\mathrm{I}(X;Y)$ and $\mathrm{H}(X|Y)$. The encoder transforms the input X into a feature representation $\hat{X}$, and the classifier maps $\hat{X}$ to a probability vector $\hat{Y}$. In this setup, the random variables $\{Y, X, \hat{X}, \hat{Y}\}$ form a Markov chain in the order shown in fig. 1 (see Yang et al. [2025], Salamah et al. [2025] for more details). Now, in section 3.1.1 and section 3.1.2, we show how this auxiliary DNN can be leveraged to derive variational estimates for $\mathrm{I}(X;Y)$ and $\mathrm{H}(X|Y)$, respectively. Then, in section 3.1.3, we train both $f_{\boldsymbol{\theta}}(\cdot)$ and $g_{\boldsymbol{\psi}}(\cdot)$ to give us the estimates.

3.1.1 Variational Estimates for $\mathrm{I}(X;Y)$

Equipped with $f_{\boldsymbol{\theta}}(\cdot)$ and classifier $g_{\boldsymbol{\psi}}(\cdot)$, in this section we propose a method to find a variational estimation for $\mathrm{I}(X;Y)$. We start by decomposing $\mathrm{I}(X;Y)$ as follows:

$$\mathrm{I}(X;Y) = \mathrm{I}(\hat{X};Y) + \mathrm{I}(Y;X|\hat{X}). \tag{3}$$

The first term on the right-hand side of eq. (3), namely $\mathrm{I}(\hat{X};Y)$, is difficult to compute directly. To overcome this difficulty, we introduce the following Proposition:

Proposition 1. Consider a linear classifier $g_{\boldsymbol{\psi}} : \{\hat{X} \to \hat{Y} | \hat{Y} = \boldsymbol{\psi}\hat{X}\}$, parameterized by $\boldsymbol{\psi} \in \mathbb{R}^{n\times m}$ with $m \geq n$. If $\boldsymbol{\psi}$ has full column rank, then

$$\mathrm{I}(\hat{X};Y) \; = \; \mathrm{I}(\hat{Y};Y). \tag{4}$$

Now, we can write

$$\mathrm{I}(\hat{Y};Y) = \mathrm{H}(Y) - \mathrm{H}(Y|\hat{Y}) \geq \mathrm{H}(Y) - \mathrm{H}(Y|\hat{Y},Y) = \mathrm{H}(Y) + \mathbb{E}_Y \log P_{Y|\hat{Y}}, \tag{5}$$

where the first inequality becomes equality if the classifier is Bayes-optimal. Now, the quantities in eq. (5) can be easily computed; specifically, $\mathrm{H}(Y)$ is simply the entropy of the ground truth distribution, which is constant for a given dataset., and $\mathbb{E}_Y \log P_{Y|\hat{Y}}$ is the average of cross-entropy of the output, In practice, we use one-hot probabilities to estimate the Bayes probabilities [Ye et al., 2024, Hamidi et al., 2024a].

The second term on the right-hand side of eq. (3), namely $\mathrm{I}(Y;X|\hat{X})$, is difficult to compute directly. In what follows, we propose to minimize this term so that $\mathrm{I}(\hat{X};Y)$ forms a tight lower bound on $\mathrm{I}(X;Y)$. To motivate this, we first present proposition 2.

Proposition 2. For an encoder $f_{\boldsymbol{\theta}}(\cdot)$ parametrized by $\boldsymbol{\theta}$

$$\min_{\boldsymbol{\theta}} \mathrm{I}(Y; X|\hat{X}) \equiv \max_{\boldsymbol{\theta}} \mathrm{I}(X; \hat{X}). \tag{6}$$

As per proposition 2, we shall train $f_{\boldsymbol{\theta}}(\cdot)$ and $g_{\boldsymbol{\psi}}(\cdot)$ to maximize $\mathrm{I}(X;\hat{X})$. The details of this training process are provided in section 3.1.3. In this manner, we find a variational estimation for $\mathrm{I}(X;Y)$ which we denote by $\underline{\mathrm{I}}(X;Y)$. Particularly,

$$\underline{\mathrm{I}}(X;Y) = \mathrm{H}(\hat{Y}) + \mathbb{E}_Y \log P_{\hat{Y}|Y}. \tag{7}$$

3.1.2 Variational Estimates for $\mathrm{H}(X|Y)$

The term $\mathrm{H}(X|Y)$ can be expanded as

$$\mathrm{H}(X|Y) = \mathrm{I}(X;\hat{X}|Y) + \mathrm{H}(X|\hat{X},Y). \tag{8}$$

To compute $\mathrm{I}(X;\hat{X}|Y)$, we introduce the following Proposition.

Proposition 3. Assume that the feature representation $\hat{X}$ has zero mean [He et al., 2015, Hinton et al., 2015]. Then, $\mathrm{I}(X;\hat{X}|Y) = \mathrm{I}(X;\sigma(\hat{X})|Y)$.

Despite $\mathrm{I}(X;\hat{X}|Y)$, the term $\mathrm{I}(X;\sigma(\hat{X})|Y)$ can indeed be calculated analytically using the same approach as used in [Yang et al., 2025, Ye et al., 2024]:

$$\mathrm{I}(X;\sigma(\hat{X})|Y) = \sum_{y\in[C]} P_Y(y)\, \mathrm{I}(X;\sigma(\hat{X})|y) = \sum_{y\in[C]} P_Y(y)\, \mathbb{E}_{X|Y}\mathrm{KL}(\sigma(\hat{X})||Q^y) \tag{9}$$

$$= \mathbb{E}_{X,Y}\mathrm{KL}(\sigma(\hat{X})||Q^Y), \tag{10}$$

where Q^y can be computed as $\frac{1}{|D_y|}\sum_{x\in D_y}\sigma(\hat{X})$ [Yang et al., 2025].

In addition, the term $\mathrm{H}(X|\hat{X},Y)$ is not easy to compute, so we introduce the following proposition to minimize it such that $\mathrm{I}(X;\hat{X}|Y)$ becomes a tight lower bound for $\mathrm{H}(X|Y)$.

Proposition 4. For an encoder $f_{\boldsymbol{\theta}}(\cdot)$ parametrized by θ

$$\min_{\theta} \mathrm{H}(X|\hat{X},Y) \equiv \max_{\theta} \mathrm{I}(X;\hat{X}). \tag{11}$$

As such, we have found a variational estimation for $\mathrm{H}(X|Y)$ which we denote by $\underline{\mathrm{H}}(X|Y)$. Particularly,

$$\underline{\mathrm{H}}(X|Y) = \mathbb{E}_{X,Y}\mathrm{KL}(\sigma(\hat{X})||Q^Y). \tag{12}$$

3.1.3 Training Variational Estimator

According to proposition 2 and proposition 4, obtaining tight lower bounds for $\mathrm{I}(X;Y)$ and $\mathrm{H}(X|Y)$, denoted as $\underline{\mathrm{I}}(X;Y)$ and $\underline{\mathrm{H}}(X|Y)$ respectively, requires training $f_{\boldsymbol{\theta}}(\cdot)$ and $g_{\boldsymbol{\psi}}(\cdot)$ to maximize $\mathrm{I}(X;\hat{X})$. We refer to this DNN, which consists of the concatenation of $f_{\boldsymbol{\theta}}(\cdot)$ and $g_{\boldsymbol{\psi}}(\cdot)$, as the variational estimator (VE) since it provides tight estimations $\underline{\mathrm{I}}(X;Y)$ and $\underline{\mathrm{H}}(X|Y)$. The training procedure for the VE is described in this Section I. To establish the theoretical foundation for this approach, we first introduce the following theorem.

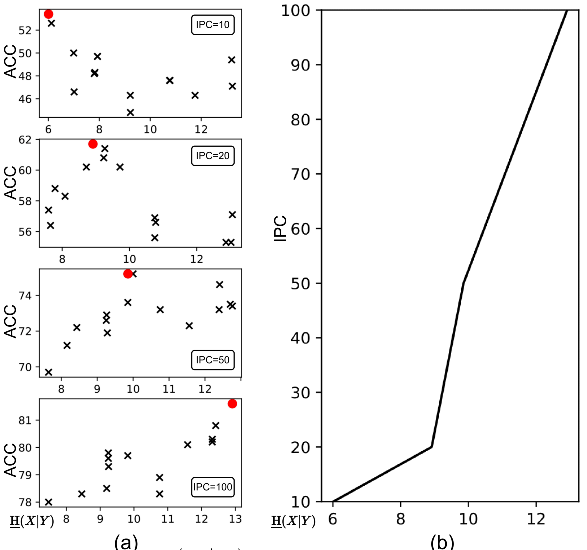

Figure 2: (a) $\underline{\mathrm{H}}(X|Y)$ of selected subsets Vs. model validation accuracy for different IPC settings; the subset associated with the highest model accuracy is marked with a red dot. (b) $\underline{\mathrm{H}}(X|Y)$ of the best subset compared across different IPC settings.

Theorem 1. Consider the mapping $\hat{X} = f_{\boldsymbol{\theta}}(X)$, where $f_{\boldsymbol{\theta}}(\cdot)$ is an encoder parametrized by $\boldsymbol{\theta}$. Then,

$$\max_{\boldsymbol{\theta}} \mathrm{I}(X;\hat{X}) \equiv \min_{\boldsymbol{\theta}} \big[\mathrm{H}(\tilde{Y}|\hat{X}) - \mathrm{I}(X;\hat{X}|\tilde{Y})\big], \tag{13}$$

where $\tilde{Y}$ is an auxiliary random variable denoting the labeling rule for each image.

As per theorem 1, to maximize $\mathrm{I}(X;\hat{X})$ one can instead maximize the three terms: $\mathrm{H}(\tilde{Y})$, $-\mathrm{H}(\tilde{Y}|\hat{X})$, and $\mathrm{I}(X;\hat{X}|\tilde{Y})$. In the following, we discuss how to maximize these three terms.

• $\mathrm{H}(\tilde{Y})$ depends only on the statistics of the random variable $\tilde{Y}$. In addition, since it is an auxiliary random variable, it can be defined such that $\mathrm{H}(\tilde{Y})$ is maximized. The following Proposition establishes a condition under which $\mathrm{H}(\tilde{Y})$ is maximized.

Proposition 5. Given a dataset $\mathcal{D}$, the entropy $\mathrm{H}(\tilde{Y})$ is maximized when each individual sample $x \in \mathcal{D}$ is associated with a unique $\tilde{y}$, and the probability distribution of $\mathrm{P}_{\tilde{Y}} = \frac{1}{|\mathcal{D}|}$, $\forall \tilde{y}$. In representation learning, this corresponds to instance discrimination.

• To maximize $-\mathrm{H}(\tilde{Y}|\hat{X})$, we first note that $\mathrm{H}(\tilde{Y}|\hat{X}) = -\sum P(\hat{x}, y)\log P(y|\hat{x})$. Since we formulate the problem as an instant discrimination, and following [Gálvez et al., 2023, Oord et al., 2018, Hamidi et al., 2024b], we approximate the conditional probability $P(y|\hat{x})$ as

$$P(\tilde{y}_i|\hat{x}_j) \approx \frac{\exp(\langle \hat{x}_i, \hat{x}_j\rangle/\tau)}{\sum_k \exp(\langle \hat{x}_i, \hat{x}_k\rangle/\tau)}, \tag{14}$$

where τ is a predetermined hyperparameter, usually referred to as the temperature [He et al., 2020]. Then,

$$\mathrm{H}(\tilde{Y}|\hat{X}) \leq \mathrm{H}(\tilde{Y}|\hat{X}, \hat{Y}|X) = -\mathbb{E}\log\left[\frac{\exp(\langle \hat{x}_i, \hat{x}_i\rangle/\tau)}{\sum_k \exp(\langle \hat{x}_i, \hat{x}_k\rangle/\tau)}\right], \tag{15}$$

where the expectation is take over $P_{\tilde{Y}|\hat{X}}$. The conditional entropy can be minimized by minimizing eq. (15).

• $\mathrm{I}(X;\hat{X}|\tilde{Y})$ can be maximized by iteratively maximizing eq. (10) and updating the $Q^y = \frac{1}{|D_y|}\sum_{x\in D_y}\sigma(\hat{X})$.

Remark 1. We acknowledge the similarity between VE and information bottleneck theory [Tishby et al., 2000, Saxe et al., 2018, Yang et al., 2024]. In section C, we elaborate on this connection and derive the objective function using an alternative approach.

3.2 Information-Guided Diffusion Sampling

Based on the discussion above, to maximize $\mathrm{I}(X;\hat{X})$ for an encoder $f_{\boldsymbol{\theta}}(\cdot)$, one can train it to maximize the following objective function:

$$\mathcal{J}_{\mathrm{VE}} = -\mathbb{E}_{\hat{X},Y}\log\left[\frac{\exp(\langle \hat{x}_i, \hat{x}_i\rangle/\tau)}{\sum_k \exp(\langle \hat{x}_i, \hat{x}_k\rangle/\tau)}\right] + \lambda\, \mathbb{E}_{\hat{X},Y}\mathrm{KL}(\sigma(\hat{X})||Q^Y), \tag{16}$$

where λ is a hyperparameter that balances the effects of two terms in the objective function. We note that VE reduces to MOCO [He et al., 2020] when $\lambda = 0$.

By maximizing $\mathcal{J}_{\mathrm{VE}}$, we can effectively train $f_{\boldsymbol{\theta}}(\cdot)$.

Once $f_{\boldsymbol{\theta}}(\cdot)$ is trained, we proceed to the next step. We freeze the parameters of $f_{\boldsymbol{\theta}}(\cdot)$ and train the classifier $g_{\boldsymbol{\psi}}(\cdot)$ using the standard cross-entropy (CE) loss. Once $f_{\boldsymbol{\theta}}(\cdot)$ and $g_{\boldsymbol{\psi}}(\cdot)$ are trained, they are fixed and used during the sampling process of the diffusion model as discussed in the next subsection.

To recap, our primary objective was to maximize the function $\mathrm{I}(X;Y) + \beta\,\mathrm{H}(X|Y)$ (see eq. (2)) during the sampling process of the diffusion model. To achieve this, we derived variational estimates for $\mathrm{I}(X;Y)$ (eq. (7)) and $\mathrm{H}(X|Y)$ (eq. (12)) by leveraging the training of a variational estimator (VE). Using these estimates, the objective function in eq. (2) is reformulated as:

$$\mathcal{L}_{IGDS} = \mathbb{E}\log P_{Y|\hat{Y}} + \beta\mathrm{I}(\hat{X};\sigma(\hat{X})|Y). \tag{17}$$

Table 1: Comparing model's performance in terms of accuracy on ImageWoof validation set. All results are evaluated under the resolution 256 × 256. We use **bold** number and asterisk (*) to denote the best and the second best results, respectively.

IPC (Ratio)	Test Model	Random	K-Center	Herding	DiT	DM	IDC-1	GLaD	MiniMax	RDED	Ours
1 (0.08%)	ConvNet-6	$14.2_{\pm0.9}$	$15.6_{\pm1.0}$	-	$12.7_{\pm0.6}$	$21.1_{\pm0.5}{}^{*}$	-	-	$15.2_{\pm0.6}$	$18.5_{\pm0.9}$	$\mathbf{23.1_{\pm0.8}}$
	ResNetAP-10	$17.8_{\pm2.4}$	$18.3_{\pm0.6}$	-	$18.0_{\pm1.3}$	-	-	-	$18.9_{\pm2.4}{}^{*}$	-	$\mathbf{23.6_{\pm0.9}}$
	ResNet-18	$13.5_{\pm0.4}$	$12.5_{\pm0.8}$	-	$15.3_{\pm0.7}$	-	-	-	$14.6_{\pm0.6}$	$20.8_{\pm1.2}{}^{*}$	$\mathbf{22.8_{\pm0.8}}$
10 (0.4%)	ConvNet-6	$24.3_{\pm1.1}$	$19.4_{\pm0.9}$	$26.7_{\pm0.5}$	$34.2_{\pm1.1}$	$26.9_{\pm1.2}$	$33.3_{\pm1.1}$	$33.8_{\pm0.9}$	$37.0_{\pm1.0}$	$40.6_{\pm2.0}{}^{*}$	$\mathbf{41.9_{\pm1.5}}$
	ResNetAP-10	$29.4_{\pm0.8}$	$22.1_{\pm0.1}$	$32.0_{\pm0.3}$	$34.7_{\pm0.5}$	$30.3_{\pm1.2}$	$39.1_{\pm0.5}$	$32.9_{\pm0.9}$	$39.2_{\pm1.3}{}^{*}$	-	$\mathbf{43.5_{\pm0.3}}$
	ResNet-18	$27.7_{\pm0.9}$	$21.1_{\pm0.4}$	$30.2_{\pm1.2}$	$34.7_{\pm0.4}$	$33.4_{\pm0.7}$	$37.3_{\pm0.2}$	$31.7_{\pm0.8}$	$37.6_{\pm0.9}$	$38.5_{\pm2.1}{}^{*}$	$\mathbf{40.7_{\pm0.5}}$
20 (1.6%)	ConvNet-6	$29.1_{\pm0.7}$	$21.5_{\pm0.8}$	$29.5_{\pm0.3}$	$36.1_{\pm0.8}$	$29.9_{\pm1.0}$	$35.5_{\pm0.8}$	-	$37.6_{\pm0.2}{}^{*}$	-	$\mathbf{45.7_{\pm0.6}}$
	ResNetAP-10	$32.7_{\pm0.4}$	$25.1_{\pm0.7}$	$34.9_{\pm0.1}$	$41.1_{\pm0.8}$	$35.2_{\pm0.6}$	$43.3_{\pm0.3}$	-	$45.8_{\pm0.5}{}^{*}$	-	$\mathbf{55.1_{\pm0.6}}$
	ResNet-18	$29.7_{\pm0.5}$	$23.6_{\pm0.3}$	$32.2_{\pm0.6}$	$40.5_{\pm0.5}$	$29.8_{\pm1.7}$	$38.6_{\pm0.2}$	-	$42.5_{\pm0.6}{}^{*}$	-	$\mathbf{49.9_{\pm0.7}}$
50 (3.8%)	ConvNet-6	$41.3_{\pm0.6}$	$36.5_{\pm1.0}$	$40.3_{\pm0.7}$	$46.5_{\pm0.8}$	$44.4_{\pm1.0}$	$43.9_{\pm1.2}$	-	$53.9_{\pm0.6}$	$61.5_{\pm0.3}{}^{*}$	$\mathbf{65.3_{\pm1.4}}$
	ResNetAP-10	$47.2_{\pm1.3}$	$40.6_{\pm0.4}$	$49.1_{\pm0.7}$	$49.3_{\pm0.2}$	$47.1_{\pm1.1}$	$48.3_{\pm1.0}$	-	$56.3_{\pm1.0}{}^{*}$	-	$\mathbf{70.2_{\pm0.8}}$
	ResNet-18	$47.9_{\pm1.8}$	$39.6_{\pm1.0}$	$48.3_{\pm1.2}$	$50.1_{\pm0.5}$	$46.2_{\pm0.6}$	$48.3_{\pm0.8}$	-	$57.1_{\pm0.6}$	$68.5_{\pm0.7}{}^{*}$	$\mathbf{71.3_{\pm0.2}}$
100 (7.7%)	ConvNet-6	$52.2_{\pm0.4}$	$45.1_{\pm0.5}$	$54.4_{\pm1.1}$	$53.4_{\pm0.3}$	$55.0_{\pm1.3}$	$53.2_{\pm0.9}$	-	$61.1_{\pm0.7}{}^{*}$	-	$\mathbf{67.2_{\pm0.2}}$
	ResNetAP-10	$59.4_{\pm1.0}$	$54.8_{\pm0.2}$	$61.7_{\pm0.9}$	$58.3_{\pm0.8}$	$56.4_{\pm0.8}$	$56.1_{\pm0.9}$	-	$64.5_{\pm0.2}{}^{*}$	-	$\mathbf{76.7_{\pm0.3}}$
	ResNet-18	$61.5_{\pm1.3}$	$50.4_{\pm0.4}$	$59.3_{\pm0.7}$	$58.9_{\pm1.3}$	$60.2_{\pm1.0}$	$58.3_{\pm1.2}$	-	$65.7_{\pm0.4}{}^{*}$	-	$\mathbf{77.3_{\pm0.7}}$

The objective function $\mathcal{L}_{IGDS}$ can be maximized during the sampling process of any diffusion model. As an example, algorithm 1 illustrates how our method can be integrated with the denoising diffusion probabilistic model (DDPM) [Ho et al., 2020]. We refer to the resulting DM-based sampling method as information-guided diffusion sampling (IGDS).

3.3 How to select β in eq. (17)

The parameter β in eq. (17) should be selected based on the IPC. To illustrate this, we generate multiple subsets of Tiny ImageNet with varying $\underline{\mathrm{H}}(X|Y)$ and IPC values. To control $\underline{\mathrm{H}}(X|Y)$, we apply a weighted sampling method, which is detailed in the Supplementary Materials. We then train ConvNet-4 on these subsets and report the classification accuracies. Finally, we plot the relationship between $\underline{\mathrm{H}}(X|Y)$ and model validation accuracy across different IPC settings in fig. 2. As observed, higher IPC settings require greater contextual detail, while lower IPC settings benefit from a stronger emphasis on prototype information

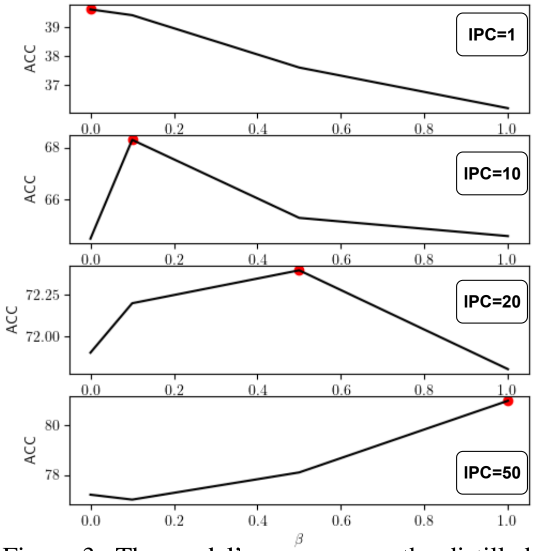

Figure 3: The model's accuracy on the distilled dataset Vs. β. As observed, lower IPC settings favor smaller β values, whereas higher IPC settings require an increased β value accordingly.

4 Experiments

• **Implementation Details of IGDS.** We adopt the pre-trained DDPM model [Dhariwal and Nichol, 2021] and use the pre-trained MoCo model [He et al., 2020] as the encoder. To smooth gradients, we follow [Ma et al., 2024] and replace the ReLU activation function with SoftPlus [Dugas et al., 2000] using $\beta = 3$. A linear classifier is then trained on top of the frozen encoder. During the DM sampling, we set the temperature to $\tau = 0.07$ and run the diffusion process for 250 steps in all experiments. Following [Sun et al., 2024], we enhance sample information by merging four images from the same class into a single composite image. We report the evaluation protocol in section J, and also section K provides sample images from the distilled datasets generated by IGDS. All experiments are conducted on a single NVIDIA V100 GPU. Full implementation details, including code and configurations, are available in our *GitHub* repository.

• **Datasets.** To evaluate the effectiveness of the proposed method, we conduct experiments mainly on several benchmark datasets. We select ImageNet-1K [Deng et al., 2009] and three well-known subsets of ImageNet: ImageNette, ImageWoof, and Tiny ImageNet [Le and Yang, 2015] . ImageNet is a large-scale visual recognition dataset containing approximately 1.2 million training images and 50,000 validation images. ImageNette, a subset of ImageNet, provides a smaller and more manageable dataset for testing deep learning models, while ImageWoof focuses on 10 dog breeds, offering a fine-grained classification task. Both ImageNette and ImageWoof use a spatial resolution of 224 × 224. Tiny ImageNet, on the other hand, is a small, balanced subset of ImageNet, with

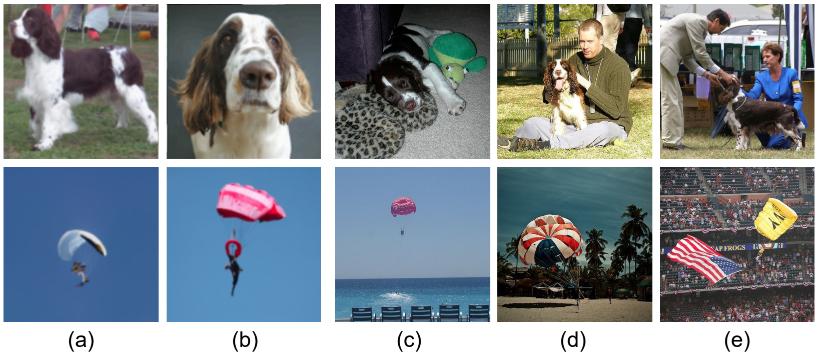

Figure 4: Illustration of prototype and contextual information. (i) Top: English Springer; (ii) Bottom: Parachute. The first two columns show synthetic images with low contextual information, while the last three columns display natural images from the same classes.

its training set consisting of 200 classes—each class containing 500 samples resized to a spatial resolution of 64×64.

• **Network Architectures.** Following previous work [Cazenavette et al., 2022, Cui et al., 2023, Guo et al., 2024], we use ConvNet-4 [LeCun et al., 1998] for the Tiny ImageNet dataset and ConvNet-6 for the ImageWoof dataset. Additionally, we employ ResNetAP-10, a variant of ResNet-10 where all pooling layers are replaced with average pooling, and ResNet-18 for all experiments.

4.1 Comparison with State-of-the-art Methods

We first report the experimental results for ImageWoof in table 1. As seen, at a low IPC setting (IPC-1), the performance of all generative-based dataset distillation methods, except IGDS, is close to that of a randomly selected subset. However, as the IPC increases, the performance gap between the baseline methods and the random selection also increases. Nevertheless, none of the baseline methods outperforms IGDS.

We defer the ImageNet-1K, ImageNette, Cifar-10, and cross-architecture results to the appendix.

4.2 Semantic Meaning of Prototype and Contextual Information

To better understand the semantic meaning of prototype and contextual information, we visualize samples generated by IGDS with $\beta = 0$, where *only* prototype information is maximized during the DM sampling process. These synthetic images are shown in the first two columns of fig. 4 (columns (a) and (b)). For direct contrast, three natural images randomly selected from the same class are displayed in the last three columns (columns (c) to (e)). As observed, the synthetic images with low contextual information feature plain backgrounds and minimal context, whereas the natural images exhibit richer contextual details.

5 Conclusions and Future Works

In this work, we addressed the limitations of diffusion model-based dataset distillation in low-IPC settings through an information-theoretic approach. We identified prototype information $\mathrm{I}(X;Y)$ and contextual information $\mathrm{H}(X|Y)$ as essential components and proposed maximizing $\mathrm{I}(X;Y) + \beta\mathrm{H}(X|Y)$ during sampling, with β adapted to IPC. To handle intractability, we introduced variational estimations using a deep neural network. Our proposed method, information-guided diffusion sampling (IGDS), seamlessly integrated with diffusion models and achieved state-of-the-art performance on Tiny ImageNet and ImageNet subsets, particularly in low-IPC regimes.

Despite the theoretical contributions and promising results of the proposed information-guided diffusion sampling (IGDS) method, this work has several limitations. First, like previous studies, we use a pretrained diffusion model as the prior distribution for natural images. While this approach is intuitive, its optimality for dataset distillation remains unverified. In addition, it restricts applicability by requiring a pretrained diffusion model for the target dataset. Second, during the IGDS process, gradients must be backpropagated through both the classifier and encoder to guide the diffusion process, increasing computational costs. We report the runtime analysis in section D. Addressing these limitations and integrating the proposed framework with more advanced priors remain important directions for future work.

Acknowledgments

This research was supported in part by JSPS KAKENHI Grant Numbers JP23K21676, JP24K02942, JP24K23849, and JP25K21218.

References

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proc. CVPR*, pages 4750–4759, 2022.

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Generalizing dataset distillation via deep generative prior. In *Proc. CVPR*, pages 3739–3748, 2023.

Mingyang Chen, Jiawei Du, Bo Huang, Yi Wang, Xiaobo Zhang, and Wei Wang. Influence-guided diffusion for dataset distillation. In *Proc. ICLR*, 2025. URL `https://openreview.net/forum?id=0whx8MhysK`.

Zhixiang Chi, Li Gu, Tao Zhong, Huan Liu, YUANHAO YU, Konstantinos N Plataniotis, and Yang Wang. Adapting to distribution shift by visual domain prompt generation. In *The Twelfth International Conference on Learning Representations*, 2024.

Zhixiang Chi, Li Gu, Huan Liu, Ziqiang Wang, Yanan Wu, Yang Wang, and Konstantinos N Plataniotis. Learning to adapt frozen clip for few-shot test-time domain adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Zhixiang Chi, Yanan Wu, Li Gu, Huan Liu, Ziqiang Wang, Yang Zhang, Yang Wang, and Konstantinos N Plataniotis. Plug-in feedback self-adaptive attention in clip for training-free open-vocabulary segmentation. *arXiv preprint arXiv:2508.20265*, 2025b.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006. ISBN 0471241954.

Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Scaling up dataset distillation to imagenet-1k with constant memory. In *Proc. ICML*, pages 6565–6590, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, pages 248–255, 2009.

Wenxiao Deng, Wenbin Li, Tianyu Ding, Lei Wang, Hongguang Zhang, Kuihua Huang, Jing Huo, and Yang Gao. Exploiting inter-sample and inter-feature relations in dataset distillation. In *Proc. CVPR*, pages 17057–17066, 2024.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Proc. NeurIPS*, pages 8780–8794, 2021.

Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. In *Proc. NeurIPS*, 2000.

Borja Rodrıguez Gálvez, Arno Blaas, Pau Rodríguez, Adam Golinski, Xavier Suau, Jason Ramapuram, Dan Busbridge, and Luca Zappella. The role of entropy and reconstruction in multi-view self-supervised learning. In *Proc. ICML*, pages 29143–29160, 2023.

Jianyang Gu, Saeed Vahidian, Vyacheslav Kungurtsev, Haonan Wang, Wei Jiang, Yang You, and Yiran Chen. Efficient dataset distillation via minimax diffusion. In *Proc. CVPR*, pages 15793–15803, 2024a.

Jianyang Gu, Saeed Vahidian, Vyacheslav Kungurtsev, Haonan Wang, Wei Jiang, Yang You, and Yiran Chen. Efficient dataset distillation via minimax diffusion. In *Proc. CVPR*, pages 15793–15803, 2024b.

Ziyao Guo, Kai Wang, George Cazenavette, Hui Li, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching. In *Proc. ICLR*, 2024.

Shayan Mohajer Hamidi and Linfeng Ye. Robustness against adversarial attacks via learning confined adversarial polytopes. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5670–5674, 2024. doi: 10.1109/ICASSP48485.2024.10446776.

Shayan Mohajer Hamidi and Linfeng Ye. Distributed quasi-newton method for fair and fast federated learning. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL `https://openreview.net/forum?id=KbteA50cni`.

Shayan Mohajer Hamidi, Xizhen Deng, Renhao Tan, Linfeng Ye, and Ahmed Hussein Salamah. How to train the teacher model for effective knowledge distillation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024a.

Shayan Mohajer Hamidi, Renhao Tan, Linfeng Ye, and En-Hui Yang. Fed-it: Addressing class imbalance in federated learning through an information- theoretic lens. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pages 1848–1853, 2024b. doi: 10.1109/ISIT57864.2024.10619204.

R. V. L. Hartley. Transmission of information. *The Bell System Technical Journal*, 7(3):535–563, 1928. doi: 10.1002/j.1538-7305.1928.tb01236.x.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. ICCV*, pages 1026–1034, 2015.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. CVPR*, pages 9729–9738, 2020.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015. URL `https://api.semanticscholar.org/CorpusID:7200347`.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proc. NeurIPS*, pages 6840–6851, 2020.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. In *Proc. ICML*, pages 11102–11118, 2022.

Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

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

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Longzhen Li, Guang Li, Ren Togo, Keisuke Maeda, Takahiro Ogawa, and Miki Haseyama. Generative dataset distillation: Balancing global structure and local details. In *Proc. CVPR Workshop*, pages 7664–7671, 2024.

Mingzhuo Li, Guang Li, Jiafeng Mao, Takahiro Ogawa, and Miki Haseyama. Diversity-driven generative dataset distillation based on diffusion model with self-adaptive memory. In *Proc. ICIP*, 2025.

Jiajun Ma, Tianyang Hu, Wenjia Wang, and Jiacheng Sun. Elucidating the design space of classifier-guided diffusion generation. In *Proc. ICLR*, 2024. URL `https://openreview.net/forum?id=9DXXMXnIGm`.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Noveen Sachdeva and Julian McAuley. Data distillation: A survey. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=lmXMXP74TO`. Survey Certification.

Ahmad Sajedi, Samir Khaki, Ehsan Amjadian, Lucy Z Liu, Yuri A Lawryshyn, and Konstantinos N Plataniotis. Datadam: Efficient dataset distillation with attention matching. In *Proc. ICCV*, pages 17097–17107, 2023.

Ahmed H. Salamah, Kaixiang Zheng, Linfeng Ye, and En-Hui Yang. Jpeg compliant compression for dnn vision. *IEEE Journal on Selected Areas in Information Theory*, 5:520–533, 2024.

Ahmed H Salamah, Shayan Mohajer Hamidi, and En-Hui Yang. A coded knowledge distillation framework for image classification based on adaptive jpeg encoding. *Pattern Recognition*, 158: 110966, 2025.

Andrew Michael Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan Daniel Tracey, and David Daniel Cox. On the information bottleneck theory of deep learning. In *Proc. ICLR*, 2018. URL `https://openreview.net/forum?id=ry_WPG-A-`.

C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3): 379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.

Duo Su, Junjie Hou, Guang Li, Ren Togo, Rui Song, Takahiro Ogawa, and Miki Haseyama. Generative dataset distillation based on diffusion model. In *Proc. ECCV Workshop*, pages 1–12, 2024.

Peng Sun, Bei Shi, Daiwei Yu, and Tao Lin. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. In *Proc. CVPR*, pages 9390–9399, 2024.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. Cafe: Learning to condense dataset by aligning features. In *Proc. CVPR*, pages 12196–12205, 2022.

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

Yanan Wu, Zhixiang Chi, Yang Wang, Konstantinos N Plataniotis, and Songhe Feng. Test-time domain adaptation by learning domain-aware batch normalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15961–15969, 2024.

En-hui Yang and Linfeng Ye. Markov knowledge distillation: Make nasty teachers trained by self-undermining knowledge distillation fully distillable. In *European Conference on Computer Vision*, pages 154–171. Springer, 2024.

En-Hui Yang, Shayan Mohajer Hamidi, Linfeng Ye, Renhao Tan, and Beverly Yang. Conditional mutual information constrained deep learning for classification (2023). *URL https://arxiv.org/abs/2309.09123*, 5.

En-Hui Yang, Shayan Mohajer Hamidi, Linfeng Ye, Renhao Tan, and Beverly Yang. Conditional mutual information constrained deep learning: Framework and preliminary results. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pages 569–574, 2024. doi: 10.1109/ISIT57864.2024.10619241.

En-Hui Yang, Shayan Mohajer Hamidi, Linfeng Ye, Renhao Tan, and Beverly Yang. Conditional mutual information constrained deep learning for classification. *IEEE Transactions on Neural Networks and Learning Systems*, 36(8):15436–15448, 2025. doi: 10.1109/TNNLS.2025.3540014.

Linfeng Ye and Shayan Mohajer Hamidi. Thundernna: a white box adversarial attack. *arXiv preprint arXiv:2111.12305*, 2021.

Linfeng Ye, En-hui Yang, and Ahmed H. Salamah. Modeling and energy analysis of adversarial perturbations in deep image classification security. In *2022 17th Canadian Workshop on Information Theory (CWIT)*, pages 62–67, 2022. doi: 10.1109/CWIT55308.2022.9817678.

Linfeng Ye, Shayan Mohajer Hamidi, Renhao Tan, and En-Hui YANG. Bayes conditional distribution estimation for knowledge distillation based on conditional mutual information. In *Proc. ICLR*, 2024. URL `https://openreview.net/forum?id=yV6wwEbtkR`.

Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. In *Proc. NeurIPS*, 2023. URL `https://openreview.net/forum?id=5Fgdk3hZpb`.

Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proc. ICCV*, pages 6022–6031, 2019.

Bo Zhao and Hakan Bilen. Synthesizing informative training samples with gan. In *Proc. NeurIPS Workshop*, pages 1–13, 2022.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *Proc. ICLR*, 2021. URL `https://openreview.net/forum?id=mSAKhLYLSsl`.

Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation. In *Proc. CVPR*, pages 7856–7865, 2023.

Tao Zhong, Zhixiang Chi, Li Gu, Yang Wang, Yuanhao Yu, and Jin Tang. Meta-dmoe: Adapting to domain shift by meta-distillation from mixture-of-experts. *Advances in Neural Information Processing Systems*, 2022.

Appendix

A Weighed Sampling to Generated Subsets with Different $\underline{\mathrm{H}}(X|Y)$ values

In this section, we describe how to perform weighted sampling to generate the subset of a dataset with different $\underline{\mathrm{H}}(X|Y)$ values. Given a dataset $\mathcal{D}$ of size n with C classes, $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, where each $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in [C]$, a pretrained encoder $f(\cdot)$ trained using the VE method on $\mathcal{D}$, and a classifier $g(\cdot)$, we first filter out all misclassified samples to ensure that the remaining samples' contextual information can be captured by $g(f(\cdot))$. In the Shannon sense, the contextual information for each sample within class y is quantified using the KL divergence $\mathrm{KL}(\sigma(x)||Q^y)$, where $\sigma(x)$ is a probability vector obtained by applying the softmax function to the feature map $\hat{x}$, and Q^y is estimated as $\frac{1}{|D_y|}\sum_{x \in D_y} H$. Given a target $\alpha_{\underline{H}(\cdot|y)}$ value, we compute the probability $\tilde{P}_{X|Y}(\cdot|y)$ of each sample being selected as follows:

$$\tilde{P}_{X|Y}(x|y) = \frac{\exp(-(\mathrm{KL}(\sigma(x)||Q^y) - \alpha_{\underline{H}(\cdot|y)})^2)}{\sum_{x' \in D^y} \exp(-(\mathrm{KL}(\sigma(x')||Q^y) - \alpha_{\underline{H}(\cdot|y)})^2)}, \tag{18}$$

then, IPC samples are drawn from each class according to the probability $\tilde{P}_{X|Y}(\cdot|y)$. We visualize the samples that map to Q^Y using the pretrained encoder $f(\cdot)$ in Figure 4 to enhance understanding of the semantic meaning of prototype information and contextual information.

B Proof of Propositions

B.1 Proof of proposition 1

Proof. We first prove that for any injective function f,

$$\mathrm{I}(X;Y) = \mathrm{I}(f(X);Y), \tag{19}$$

To do so, we begin by expanding the mutual information $\mathrm{I}(\cdot;\cdot)$, and introducing the variable $Z = f(X)$:

$$\begin{aligned}\mathrm{I}(X;Y) - \mathrm{I}(f(X);Y) &= \mathrm{H}(Y|f(X)) - \mathrm{H}(Y|X) \qquad (20)\\ &= \mathrm{H}(X|f(X)). \qquad (21)\end{aligned}$$

Since f is injective, for any output $z = f(X)$, there exists a unique x such that $f(x) = z$. Therefore,

$$P(X = x|f(X) = z) = \begin{cases} 1 & \text{if } z = f(x), \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

The conditional entropy is then given by:

$$\begin{aligned}\mathrm{H}(X|f(X)) &= \mathbb{E}_{f(X)}[\mathrm{H}(X|f(X) = z)] \qquad (23)\\ &= \mathbb{E}_{f(X)}0 \qquad (24)\\ &= 0. \qquad (25)\end{aligned}$$

Thus, if f is injective, we conclude that $I(Y, f(X)) = I(Y, X)$.

Next, we show that for a matrix $\theta \in \mathbb{R}^{m,n}, m \geq n$, if θ has full column rank, then the linear mapping $\{\hat{X} \to \hat{Y}$, where $\hat{Y} = \theta\hat{X}\}$, is injective.

A function is injective if:

$$\theta x_1 = \theta x_2 \Rightarrow x_1 = x_2, \tag{26}$$

Rearranging, introducing $v = x_1 - x_2$ we get: $\theta v = 0$. For injectivity, we must show that the only solution to $\theta v = 0$ is $v = 0$.

The set of all solutions to $\theta v = 0$ is the null space of θ, denoted as:

$$Null(\theta) = \{v \in \mathbb{R}^n | \theta v = 0\}. \tag{27}$$

If the linear mapping is injective, the only vector in the null space must be the zero vector, *i.e.* $Null(\theta) = \{0\}$, which means θ has full column rank. □

We verify that the classifier's matrix after training has full column rank.

B.2 Proof of proposition 2

$\min_\theta \mathrm{I}(Y;X|\hat{X}) \equiv \max_\theta \mathrm{I}(X;\hat{X})$

Proof.

$$\begin{aligned}\mathrm{I}(Y;X|\hat{X}) =& \mathrm{I}(Y;X|\hat{X}) && (28)\\ =& \mathrm{H}(X|\hat{X}) - \mathrm{H}(X|Y,\hat{X}) && (29)\\ =& \mathrm{H}(X) - \mathrm{H}(X|Y) + \\ & \mathrm{H}(X|\hat{X}) - \mathrm{H}(X) && (30)\\ =& \mathrm{I}(X;Y) - \mathrm{I}(X;\hat{X}), && (31)\end{aligned}$$

where $\mathrm{I}(X;Y)$ is a constant, which only depends on the nature of the sampling process, *i.e.* how the dataset is collected and constructed. □

B.3 Proof of Theorem 1

Proof. Using the chain rule, the mutual information
$\mathrm{I}(X;\tilde{Y},\hat{X})$ can be expanded in two different ways:

$$\mathrm{I}(X;\tilde{Y},\hat{X}) = \mathrm{I}(X;\tilde{Y}|\hat{X}) + \mathrm{I}(X;\hat{X}) \tag{32a}$$

$$\mathrm{I}(X;\tilde{Y},\hat{X}) = \mathrm{I}(X;\hat{X}|\tilde{Y}) + \mathrm{I}(X;\tilde{Y}). \tag{32b}$$

By setting the right hand side of eq. (32a) and eq. (32b) equal to each other, we obtain

$$\begin{aligned}\mathrm{I}(X;\hat{X}) &= \mathrm{I}(X;\tilde{Y}) - \mathrm{I}(X;\tilde{Y}|\hat{X}) + \mathrm{I}(X;\hat{X}|\tilde{Y}) && (33)\\ &= \mathrm{I}(X;\hat{X};\tilde{Y}) + \mathrm{I}(X;\hat{X}|\tilde{Y}) && (34)\\ &= \mathrm{I}(\hat{X};\tilde{Y}) + \mathrm{I}(X;\hat{X}|\tilde{Y}) && (35)\\ &= \mathrm{H}(\tilde{Y}) - \mathrm{H}(\tilde{Y}|\hat{X}) + \mathrm{I}(X;\hat{X}|\tilde{Y}), && (36)\end{aligned}$$

where eq. (34) follows from the definition of interaction information, and eq. (35) holds because $\tilde{Y} \to X \to \hat{X}$ forms a Markov chain. □

B.4 Proof of proposition 3

Assume that the feature representation $\hat{X}$ has zero mean. Then, $\mathrm{I}(X;\hat{X}|Y) = \mathrm{I}(X;\sigma(\hat{X})|Y)$.

Proof. The softmax function for an input vector $x \in \mathbb{R}^N$ is define as

$$softmax(x)[i] = \frac{e^{x[i]}}{\sum_{j\in[N]} e^{x[j]}}. \tag{37}$$

Following the proof of proposition 1 in section B.1, we aim to show that the softmax function is injective if its domain is in the subspace with zero mean. Assume two vectors $x;y \in \mathbb{R}^N$, such that

$$\sum_{i\in[N]} x_i = 0, \quad \sum_{i\in[N]} y_i = 0, \tag{38}$$

$$\frac{e^{x[i]}}{\sum_{j\in[N]} e^{x[j]}} = \frac{e^{y[i]}}{\sum_{j\in[N]} e^{y[j]}}. \tag{39}$$

Rewriting eq. (39),

$$e^{x[i]} \sum_{j\in[N]} e^{y[j]} = e^{y[i]} \sum_{j\in[N]} e^{x[j]}, \ \forall i \in [N]. \tag{40}$$

We define the ratio of $\sum_{j\in[N]} e^{x[j]}$ and $\sum_{j\in[N]} e^{y[j]}$ as:

$$\Theta = \frac{\sum_{j\in[N]} e^{x[j]}}{\sum_{j\in[N]} e^{y[j]}}. \tag{41}$$

Substitute the eq. (41) into eq. (40) and take logarithm on both sides:

$$x_i = \log\Theta + y_i. \tag{42}$$

Since both x and y have zero mean, we have:

$$\sum_{i\in[N]} x_i = \sum_{i\in[N]} \log\Theta + y_i = 0, \tag{43}$$

$$N\log\Theta = 0,\ \Theta = 1. \tag{44}$$

Thus $x_i = y_i,\ \forall i \in [N]$. □

B.5 Proof of proposition 4

For a encoder f parametrized by θ.

$\min_\theta \mathrm{H}(X|\hat{X}, Y) \equiv \max_\theta \mathrm{I}(X; \hat{X})$

Proof.

$$\mathrm{I}(X;\hat{X}) = \mathrm{H}(X) - \mathrm{H}(X|\hat{X}) \tag{45}$$

$$= \mathrm{H}(X) - \mathrm{H}(X|\hat{X}, Y) \tag{46}$$

where $\mathrm{H}(X)$ is a constant, which equals the amount of information in the dataset, eq. (46) is due to $Y \to X \to \hat{X}$ forms a Markov chain. □

B.6 Proof of proposition 5

Proof. Consider a random variable Y, with N classes. Its entropy is given by

$$\mathrm{H}(Y) = -\sum_{n\in[N]} P_n \log P_n. \tag{47}$$

Without losing generality, suppose we split the first sample point y_1 into two sample points $\hat{y}_a$ and $\hat{y}_b$, such that

$$P[y_1] = P[\hat{y}_a] + P[\hat{y}_b], \tag{48}$$

$$s.t.\ P[\hat{y}_a] > 0; P[\hat{y}_b] > 0, \tag{49}$$

This transformation produces a new random variable $\hat{Y}$ with $N+1$ sample points. The change in entropy is then

$$\mathrm{H}(\hat{Y}) - \mathrm{H}(Y) \tag{50}$$

$$= -P[\hat{y}_b]\log P[\hat{y}_b] - P[\hat{y}_a]\log P[\hat{y}_a] + P[y_1]\log P[y_1] \tag{51}$$

$$= -P[\hat{y}_b]\log P[\hat{y}_b] - P[\hat{y}_a]\log P[\hat{y}_a] + \{P[\hat{y}_b] + P[\hat{y}_a]\}\log\{P[\hat{y}_b] + P[\hat{y}_a]\} \tag{52}$$

$$= -P[\hat{y}_b]\big\{\log P[\hat{y}_b] - \log\{P[\hat{y}_b] + P[\hat{y}_a]\}\big\} - P[\hat{y}_a]\big\{\log P[\hat{y}_a] - \log\{P[\hat{y}_b] + P[\hat{y}_a]\}\big\} \tag{53}$$

$$>0. \tag{54}$$

Thus $\mathrm{H}(\hat{Y}) > \mathrm{H}(Y)$, that is, splitting a single sample point into two distinct points increases the entropy of a random variable. In other words, given a set of samples, the entropy $\mathrm{H}(Y)$ is maximized when each sample is assigned to a unique class. Conversely, for a fixed number of sample points, the entropy is maximized when their probabilities are uniformly distributed. Therefore, the entropy $\mathrm{H}(Y)$ is maximized if the problem is formulated as instance discrimination. □

C Further discussion on VE and information bottleneck

In this section, we discuss the relationship between the maximized mutual information method and information bottleneck, and provide an alternative approach to derive the objective function for the VE method.

C.1 Relationship Between VE and Information Bottleneck

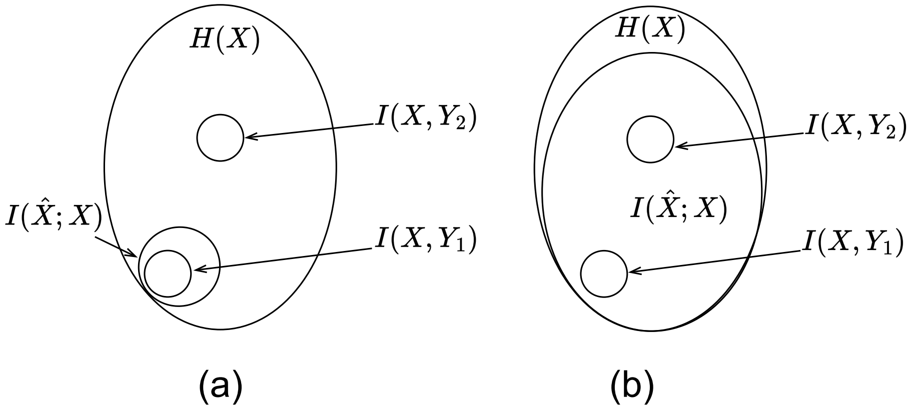

Figure 5: Mutual information between $\hat{X}$ and X for models trained with objectives of (a) information bottleneck and (b) VE (ours).

We depict the Venn diagram of which show the relationships between the $\mathrm{H}(X)$, prototype information $\mathrm{I}(X;Y)$, contextual information $\mathrm{H}(X|Y)$ and $\mathrm{I}(\hat{X};X)$ by the encoder trained by information bottleneck (a) and VE (b) in fig. 5.

Information bottleneck aims to minimize the following objective function:

$$\min \mathrm{I}(X;\hat{X}) - \beta \mathrm{I}(\hat{X};Y), \tag{55}$$

which can be interoperated as finding a compressed representation $\hat{X}$ of X that retains as much information about Y as possible, while minimizing the information retained from X.

While the target of VE differs from the information bottleneck, as it aims to maximize the mutual information $\mathrm{I}(X;\hat{X})$, as though, the compressed representation $\hat{X}$ retains as much information about X as possible.

C.2 Alternative Approach to Simplify the Prototype Information

With a slight abuse of notation, in this section, we refer to $\hat{Y}$ as the label predicted from the feature $\hat{X}$.

$$
\begin{aligned}
\mathrm{I}(\hat{X};Y) &= \mathrm{H}(Y) - \mathrm{H}(Y|\hat{X}) && (56)\\
&\geq \mathrm{H}(Y) - \mathrm{H}(E) - P(E)\log(|Y|-1) && (57)\\
&\geq \mathrm{H}(Y) - \mathrm{H}(E) - \log(|Y|-1)\mathbb{E}_X\left[1 - \sum_{i=1}^{|Y|} P_{Y|X}(i|x)P_{\hat{Y}|X}(i|x)\right] && (58)\\
&= \mathrm{H}(Y) - \mathrm{H}(E) - \log(|Y|-1)\mathbb{E}_X\left[\sum_{i=1}^{|Y|} P_{Y|X}(i|x)\Big[1 - P_{\hat{Y}|X}(i|x)\Big]\right] && (59)\\
&\geq \mathrm{H}(Y) - \mathrm{H}(E) - \log(|Y|-1)\mathbb{E}_X\left[\sum_{i=1}^{|Y|} -P_{Y|X}(i|x)\log P_{\hat{Y}|X}(i|x)\right] && (60)\\
&= \mathrm{H}(Y) - \mathrm{H}(E) - \log(|Y|-1)\mathbb{E}_X[\mathrm{H}(P_{Y|X}(i|x), P_{\hat{Y}|X}(i|x))], && (61)
\end{aligned}
$$

where eq. (57) follows from [Hartley, 1928], $\log(|Y|-1)$ is a constant, and $\mathrm{H}(E)$ approaches zero when no prediction error occurs.

D Running Time

In this section, we report the running time of the IGDS algorithm and compare its efficiency with the MiniMax [Gu et al., 2024b]. To this end, we sampled 100 images with resolution 256×256 for the ImageWoof and ImageNette datasets using both methods on the clusters we used; all comparisons are done with one single NVIDIA V100 GPU. The running time is reported in the table 2.

Table 2: The running time of MiniMax Diffusion and IGDS.

Dataset	ImageNette	ImageWoof
MiniMax	44 mins	46 mins
IGDS	57 mins	53 mins

Compared to MiniMax diffusion, IGDS slightly increases the time complexity, primarily due to the additional VE model required in the distillation process.

E Combining with Priors Beyond DDPM

An important advantage of our approach is its flexibility to integrate with a wide range of generative priors beyond DDPM. To illustrate this capability, we conducted experiments combining our method with the Minimax-DIT prior, which is recognized as a more advanced prior often leading to stronger performance. As shown in table 3, when using the Minimax-DIT prior, our method achieves an accuracy of 48.6% on the ImageWoof dataset under IPC-10. This result demonstrates that our approach not only remains effective when paired with stronger priors but can also be seamlessly combined with alternative generative models to further enhance performance.

Table 3: IGD vs. ours when using Minimax-DIT as prior.

IPC	1	10	50	100
Minimax-IGD	-	47.2	65.0	71.5
Minimax-Ours	37.6	**48.6**	**65.6**	**75.3**

F Cross-architecture performance

To evaluate the robustness of our distilled datasets across diverse model families, we follow the IGD protocol and measure performance when training four different architectures on the same distilled coresets. table 4 summarizes the test accuracies achieved by Minimax-IGD and our method under the IPC-10 setting.

Table 4: Cross-architecture performance

Method	ResNet-101	MobileNet-V2	EfficientNet-B0	SwinT
Minimax-IGD	53.4	39.7	**48.5**	44.8
Minimax-ours	**53.6**	**39.9**	48.3	**45.9**

G CIFAR-10 Results

We evaluated the performance of our approach and baseline methods on the CIFAR-10 dataset using a ResNet-18, as shown in the Tab. 5. As observed, our method achieves comparable or superior performance to previous state-of-the-art methods on this low-resolution dataset.

Table 5: Performance comparison over ResNet-18 on CIFAR-10

IPC	Test Model	SRe^2L	RDED	DIT-IGD	Ours
10	ResNet-18	29.3	**37.1**	35.8	37.0
50	ResNet-18	45.0	62.1	63.5	**64.9**

H ImageNette Results

We also present the experimental results for ImageNette and Tiny ImageNet in table 7 and table 6, respectively. The results follow a similar trend to those on the ImageWoof dataset.

I Pseudo-code

Algorithm 1 Pseudo-code of IGDS

```
Input: The number of iterations $N$, $\boldsymbol{y}$, noise levels $\{\tilde{\sigma}\}$, pre-trained encoder $f(\cdot)$ and classifier $g(\cdot)$, $\eta > 0$, $\beta > 0$, $\tau > 0$, IPC n, target label $y$.
$\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = N-1, N-2, \ldots, 0$ do
    $\hat{\boldsymbol{s}} \leftarrow \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$
    $\tilde{\boldsymbol{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}_t + (1-\bar{\alpha}_t)\hat{\boldsymbol{s}})$
    $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
    $\boldsymbol{x}'_{t-1} \leftarrow \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\tilde{\boldsymbol{x}}_0 + \tilde{\sigma}_t\boldsymbol{z}$.
    $\hat{\boldsymbol{x}}_{t-1} = f(\boldsymbol{x}'_{t-1})$
    $H_{\hat{\boldsymbol{x}}_{t-1}} = SoftMax(\boldsymbol{x}_{t-1}/\tau)$
    $Q_{t-1} = \frac{1}{n}\sum H_{\hat{\boldsymbol{x}}_{t-1}}$
    $\hat{\boldsymbol{y}}_{t-1} = g(\hat{\boldsymbol{x}}_{t-1})$
    $\mathcal{L}_{IGDS} = \mathbb{E}\log P_{\hat{y}|y} + \mathrm{H}(\hat{y}) + \beta\mathrm{KL}(H_{\hat{\boldsymbol{x}}_{t-1}}||Q_{t-1})$
    $\boldsymbol{x}_{t-1} \leftarrow \boldsymbol{x}'_{t-1} + \eta\nabla_{\boldsymbol{x}_t}\mathcal{L}_{IGDS}$.
end for
Output: $\boldsymbol{x}_0$
```

Table 7: Comparing model's performance in terms of accuracy on ImageNette validation set. All results are evaluated under the resolution 256×256. We use **bold** number and asterisk (*) to denote the best and the second best results, respectively.

Model	ResNetAP-10					ResNet-18		
IPC	Random	DiT	DM	MiniMax	Ours	RDED	SRe^2L	Ours
1	$26.7_{\pm 1.0}$	$27.3_{\pm 0.9}$	-	$30.5_{\pm 0.8}^{*}$	$\mathbf{39.6}_{\pm \mathbf{1.3}}$	$35.8_{\pm 1.0}^{*}$	$19.1_{\pm 1.1}$	$\mathbf{35.9}_{\pm \mathbf{0.7}}$
10	$54.3_{\pm 1.6}$	$59.1_{\pm 0.7}$	$60.8_{\pm 0.6}$	$62.0_{\pm 0.2}^{*}$	$\mathbf{68.3}_{\pm \mathbf{0.2}}$	$61.4_{\pm 0.4}^{*}$	$29.4_{\pm 3.0}$	$\mathbf{64.3}_{\pm \mathbf{0.6}}$
20	$63.5_{\pm 0.5}$	$64.8_{\pm 1.2}$	$66.5_{\pm 1.1}$	$66.8_{\pm 0.4}^{*}$	$\mathbf{72.4}_{\pm \mathbf{0.7}}$	-	-	$\mathbf{70.9}_{\pm \mathbf{0.3}}$
50	$76.1_{\pm 1.1}$	$73.3_{\pm 0.9}$	$76.2_{\pm 0.4}$	$76.6_{\pm 0.2}^{*}$	$\mathbf{81.0}_{\pm \mathbf{0.5}}$	$80.4_{\pm 0.5}^{*}$	$40.9_{\pm 0.3}$	$\mathbf{81.2}_{\pm \mathbf{0.4}}$

Algorithm 2 Pseudo-code for Training the $f_{\boldsymbol{\theta}}(\cdot)$

```
Input: $f_{\boldsymbol{\theta}}, f_{\boldsymbol{m}}$: initialized encoder and momentum encoder, $queue$: dictionary as a queue of $K$ keys, $m$: momentum, $aug$: random augmentation method, $\tau$: temperature and $\lambda > 0$.
$f_{\boldsymbol{\theta}}$.params = $f_{\boldsymbol{m}}$.params
for $x \in D$ do
    $x_q, x_k$ = aug($x$), aug($x$)
    $q, k = f_{\boldsymbol{\theta}}(x_q), f_{\boldsymbol{m}}(x_k)$.detach()
    $H_q, H_k$ = softmax($q$), softmax($k$)
    $Q = (H_q + H_k)/2$
    $l_{pos}, l_{neg} = \langle q, k \rangle, q@k^T$
    logits = cat([$l_{pos}, l_{neg}$], dim=1)
    labels = zeros(N)
    loss = CE (logits / $\tau$, labels) - $\lambda \mathrm{KL}(H_q || Q^Y)$
    loss.backward()
    update($f_{\boldsymbol{\theta}}$.params)
    $f_{\boldsymbol{m}}$.params = m × $f_{\boldsymbol{m}}$.params + (1-m) × $f_{\boldsymbol{\theta}}$.params
    enqueue(queue, k)
    dequeue(queue)
end for
Output: $f_{\boldsymbol{\theta}}$
```

J Evaluation Protocol

We report the evaluation protocol in this section. Three commonly used network architectures are used for evaluation:

• **ConvNet-6**, a 6-layer convolutional network, is an extension of ConvNet-3, which is commonly used in previous dataset distillation (DD) works for small-resolution images. To accommodate full-sized 256×256 ImageNet data, we add three additional layers. Each layer contains 128 feature channels, and instance normalization is applied.

• **ResNetAP-10** is a 10-layer ResNet variant in which the standard strided convolution is replaced with average pooling for downsampling, allowing for smoother feature aggregation.

Table 6: Comparing model's performance in terms of accuracy on Tiny ImageNet validation set. All results are evaluated under the resolution 64×64. We use **bold** number and asterisk (*) to denote the best and the second best results, respectively.

Model	ConvNet-4			
IPC	Random	IDM [Zhao et al., 2023]	RDED [Sun et al., 2024]	Ours
1	$6.7_{\pm 0.4}$	$10.1_{\pm 0.2}$	$\mathbf{12.0_{\pm 0.1}}$	$11.9_{\pm 0.3}{}^*$
10	$17.6_{\pm 0.3}$	$21.9_{\pm 0.3}$	$39.6_{\pm 0.1}{}^*$	$\mathbf{40.7_{\pm 0.3}}$
50	$22.4_{\pm 0.2}$	$27.7_{\pm 0.3}$	$47.6_{\pm 0.2}{}^*$	$\mathbf{50.3_{\pm 0.2}}$
Model	ResNet-18			
IPC	Random	SRe^2L [Yin et al., 2023]	RDED [Sun et al., 2024]	Ours
1	$2.2_{\pm 0.4}$	$2.6_{\pm 0.1}$	$9.7_{\pm 0.4}{}^*$	$\mathbf{9.8_{\pm 0.4}}$
10	$14.6_{\pm 0.2}$	$16.1_{\pm 0.2}$	$\mathbf{41.9_{\pm 0.2}}$	$41.2_{\pm 0.1}{}^*$
50	$35.6_{\pm 0.3}$	$41.1_{\pm 0.4}$	$58.2_{\pm 0.1}{}^*$	$\mathbf{60.1_{\pm 0.5}}$

• **ResNet-18** is an 18-layer ResNet modified to use instance normalization (IN) instead of batch normalization. Since IN performs better than batch normalization under our experimental protocol, we adopt it consistently across all ResNet-18 models.

During the evaluation training, we closely follow the protocols established in [Kim et al., 2022, Gu et al., 2024b, Ye and Hamidi, 2021]. Specifically, we use the Adam optimizer with a fixed learning rate of 0.01 across all experiments to ensure consistency in optimization. The number of training epochs for different IPC settings is detailed in Table 8. The applied data augmentations are random resize-crop, random horizontal flip, and CutMix [Yun et al., 2019].

Table 8: Evaluation training epochs across different IPC settings.

IPC	1	10	20	50	100
Epochs	2000	2000	1500	1500	1000

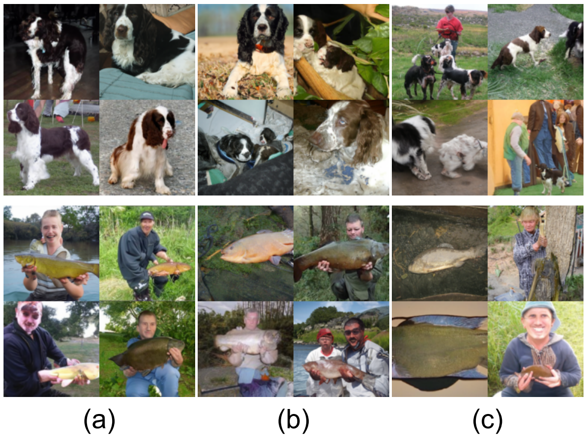

Figure 6: Generated images of two classes: English Springer (first row) and Tench (second row), with varying β values. Columns (a), (b), and (c) correspond to $\beta = \{0, 0.1, 0.5\}$, respectively.

J.1 Ablation Study on β in IGDS

In this section, we study the impact of β on IGDS, the performance of the distilled dataset under different IPC settings, and its effect on the generated images. To this end, we first examine how β influences the semantic meaning of generated images, as shown in fig. 6. Specifically, we generate 24 synthetic images for the classes English Springer and Tench, displayed in the first and second rows of fig. 6, respectively. Columns (a), (b), and (c) correspond to $\beta = \{0, 0.1, 0.5\}$. When $\beta = 0$, the generated images contain minimal contextual information. For example, the English Springer images primarily depict the dog itself, while the Tench images consistently depict a person holding the fish. As β increases, more contextual elements are incorporated into the generated images, leading to a greater diversity in semantic meaning. This effect is particularly noticeable in the English Springer images, where the background becomes richer compared to those generated with $\beta = 0$. A similar trend can be observed in the Tench images, where additional contextual details emerge as β increases. More images generated by IGDS with different β values are presented in section K.

The optimal value of β should be empirically determined for different IPC settings. To illustrate this, fig. 3 shows the test accuracy of the model as a function of β under varying IPC values. As observed, higher IPC settings benefit from a larger β, aligning with the findings discussed in section 3.3.

K Samples Generated by IGDS

In this section, we provide additional examples of distilled datasets for ImageNette and ImageWoof with IPC 10, shown in fig. 7 and fig. 8, respectively.

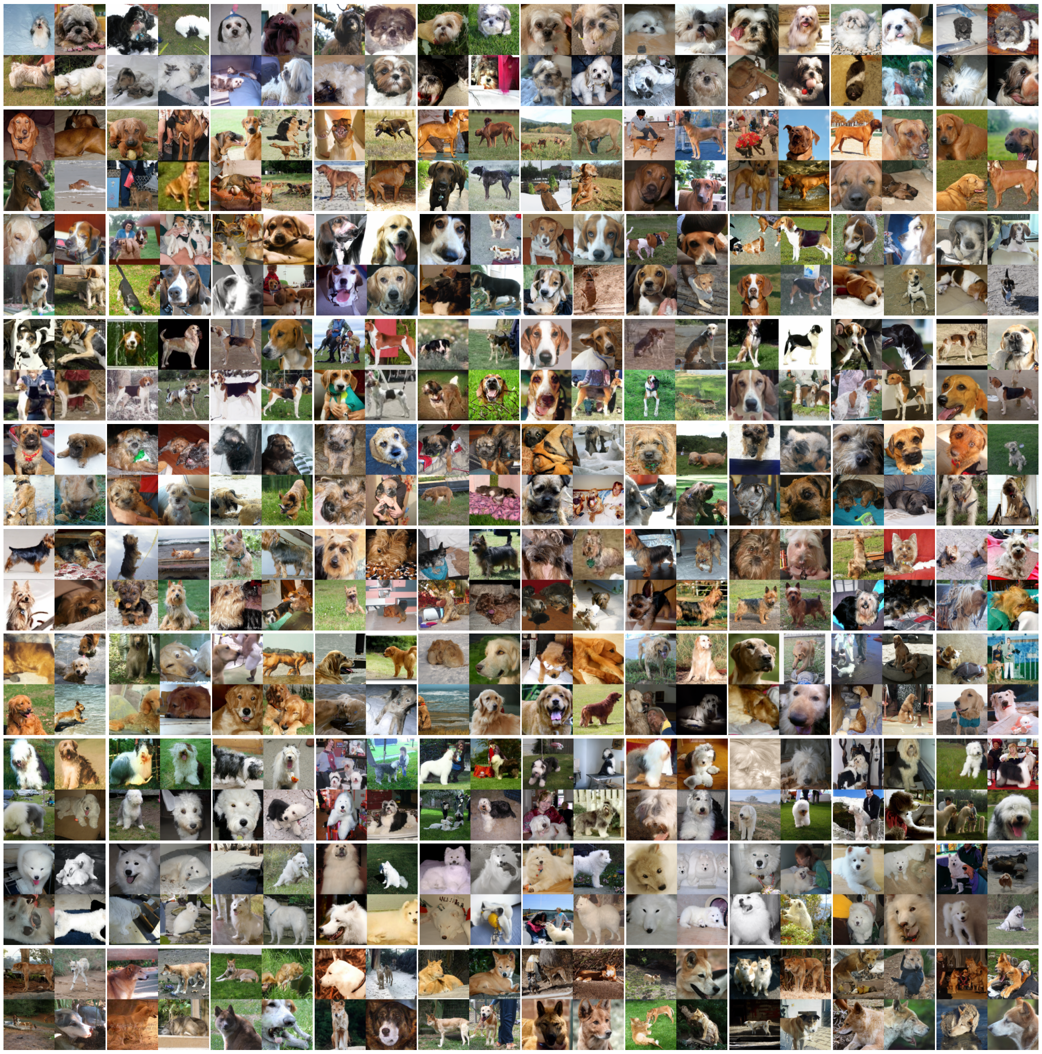

Figure 7: Distilled Image Visualization: ImageNette with IPC 10.

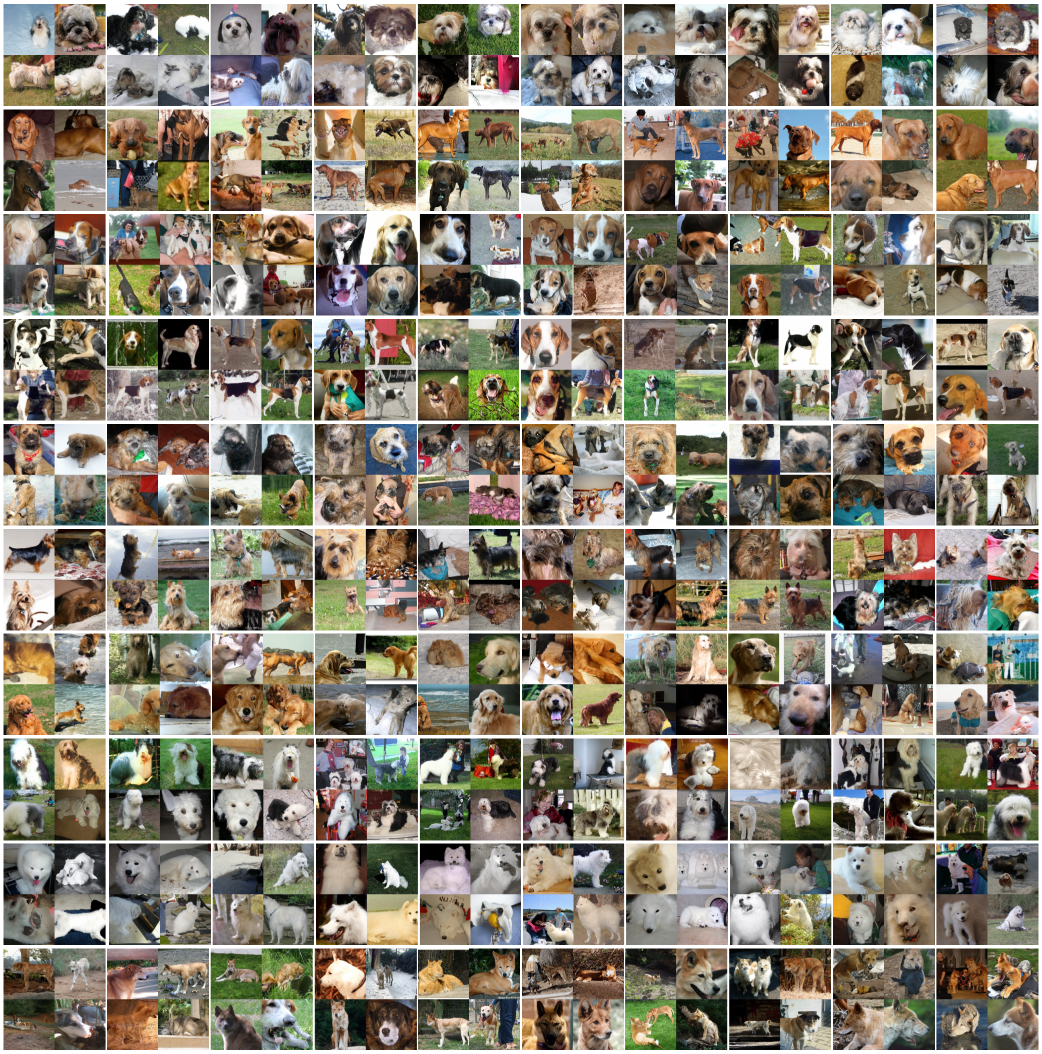

Figure 8: Distilled Image Visualization: ImageWoof with IPC 10.