Latent Probabilistic Dataset Distillation with Theoretical Guarantees

Anonymous Authors[1]

Abstract

Dataset distillation involves the compression of large datasets into smaller *coresets* that sustain similar performance to the full dataset when downstream models are trained on them – thus hugely simplifying the training task in terms of storage and computation. The current state-of-the-art methods utilize *Kernel Inducing Points* (KIP), which exploits the link between Kernel Regression and the Neural Tangent Kernel (NTK) to learn synthetic coresets that mimic the performance of a neural network on the full size, via a frequentist adaptation of the inducing point method for Gaussian processes. The frequentist regime prohibits the potential benefits of a Bayesian analysis of bounds on the number of inducing points required. The nature of the mean-squared loss employed does not lend itself to a probabilistic interpretation, while the algorithm itself is computationally intensive, as these they operated directly in the space of the data. To this end, we introduce a new variational Gaussian process-based algorithm for fast, scalable dataset distillation by learning inducing points and soft targets in the latent space of pre-trained autoencoders. Via recent observations on the similarity of the Reproducing Kernel Hilbert Space (RKHS) of the Laplace kernel and the NTK, we also develop associated guarantees on the size and efficacy of coresets over d-dimensional datasets normalized to the unit hypersphere S^{d-1}, by showing that we can get vanishingly small KL Divergence with a polynomially bound subset of the size of the data. Our method achieves competitive performance to state-of-the-art algorithms in only a fraction of the time required, often in less than one minute.

1. Introduction

First introduced by Wang et al. (2018), dataset distillation (DD) aims at extracting the knowledge of the entire training dataset into a few synthetic, distilled datapoints. The models trained on these distilled datapoints achieve high performance relative to models trained on the original, large training datasets. DD is a sensible choice for fast, cost-effective, and lightweight training of neural network models. Various applications of DD include continual learning (Liu et al., 2020; Rosasco et al., 2021; Sangermano et al., 2022; Wiewel & Yang, 2021; Masarczyk & Tautkute, 2020), neural architecture search (Zhao & Bilen, 2021; Zhao et al., 2021; Zhao & Bilen, 2023), and more.

Depending on the similarity metrics chosen for judging how close the distilled datasets are to the original datasets, different formulations of the DD problem have been proposed. For instance, Zhao et al. (2021) formulate it as a gradient matching problem between the gradients of deep neural network weights trained on the original and distilled data. Nguyen et al. (2021a; b) formulate it as a kernel ridge regression problem where the distilled data correspond to the kernel inducing points (KIP). Regardless of the formulation, DD techniques are rapidly improving, and their application domains are widening.

Among the many application domains, Nguyen et al. (2021a) claim that DD is also useful for privacy-preserving dataset creation by showing that distilled images with 90% of their pixels corrupted still exhibit limited test accuracy degradation. Although the distorted images are not humanly discernible, this illustration lacks a formal privacy definition. Dong et al. (2022) attempted to connect DD with differential privacy (Dwork Roth, 2014) based on DD's empirical robustness against known attacks. However, the empirical evaluation and theoretical analysis of their method have significant flaws, as discussed in Carlini et al. (2022).

For a provable privacy guarantee, Chen et al. (2022) applied DP-SGD, an off-the-shelf differential privacy algorithm (Abadi et al., 2016), to optimize a gradient matching objective and estimate a differentially private distilled dataset. More recently, Zheng & Li (2023) proposed a differentially private distribution matching framework, further improving the performance of Chen et al. (2022).

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

The current state-of-the-art in dataset distillation, *Kernel Inducing Points* (KIP) (Nguyen et al., 2021a; 2021b), exploits the link between Kernel Ridge Regression (KRR) and the Neural Tangent Kernel (NTK) to learn synthetic coresets that mimic the performance of a neural network trained on the full dataset. However, KIP suffers from issues in scalability, particularly due to the computational complexity of the NTK matrix over the coresets at each optimization step.

Additionally, the mean-squared loss employed in KIP does not provide a probabilistic interpretation, which is a limitation given the potential benefits of leveraging probabilistic frameworks. Recent developments in fast, finite-width NTKs suggest promising avenues for overcoming these challenges (Zheng & Li, 2023).

In this paper, we propose and analyze two new algorithms, **GPD**, or GAUSSIAN PROCESS DISTILLATION, wherein we utilize a variational Gaussian process to learn the coreset in the space of the data, and **LD**, or LATENT DISTILLATION, wherein we extend **GPD** to learning the coreset in the latent space. Our main findings and contributions are as follows:

1. We address the interpretability issues of KIP by introducing a theoretically-grounded variational method that minimizes the KL divergence between the true posterior of the surrogate Gaussian process on the whole dataset and the approximate posterior on the set of inducing points via the GAUSSIAN PROCESS DISTILLATION ALGORITHM (GPD).

2. We introduce the LATENT DISTILLATION ALGORITHM, which learns the distilled dataset in the latent space, which is vastly more scalable than current methods – at times reducing the dimensionality of the optimization problem from 784 to 32 dimensions with comparable performance. For classification tasks, we show that using the *soft labels* produced by the Gaussian process inference on the coreset can serve as superior labels for downstream training of Neural Networks.

3. Our method provides associated guarantees on the size and efficacy of coresets over d-dimensional datasets normalized to the unit hypersphere S^{d-1}.

4. We leverage the similarity between the Reproducing Kernel Hilbert Space (RKHS) of the Laplace kernel and the NTK to achieve vanishingly small KL Divergence with a polynomial bound on the size of the data, which via the equivalence of KRR with GP Regression, offers a first theoretical treatment of Dataset Distillation. We also show that coresets produced via our algorithm are differentially private.

2. Related Work

Dataset Distillation. Dataset distillation methods aim to summarize large datasets into significantly smaller datasets that still accurately represent the full dataset for downstream tasks (Jubran et al., 2019). These smaller datasets are beneficial for speeding up model training (Mirzasoleiman et al., 2020), reducing catastrophic forgetting (Aljundi et al., 2019; Rebuffi et al., 2017; Borsos et al., 2020), and enhancing interpretability (Kim et al., 2016; Bien Tibshirani, 2011). Recent work has focused on generating synthetic data points rather than selecting representative data points from the dataset (Wang et al., 2018; Bohdal et al., 2020; Sucholutsky Schonlau, 2019; Zhao et al., 2021; Zhao Bilen, 2021b; Nguyen et al., 2021b), leveraging continuous gradient-based optimization techniques.

Neural Tangent Kernels. NTKs have been extensively studied for their ability to provide exact solutions to the training dynamics of infinitely-wide neural networks (Jacot et al., 2018). They offer a powerful tool for understanding and optimizing neural networks' training processes and have been applied in various settings, including dataset distillation (Nguyen et al., 2021a; b).

Kernel Inducing Points (KIP). KIP (Nguyen et al., 2021a; 2021b) connects Kernel Ridge Regression (KRR) and NTKs to learn synthetic coresets. This method, however, faces scalability issues due to the computation of the NTK matrix at each optimization step. Moreover, the mean-squared loss employed lacks a probabilistic interpretation.

Variational Gaussian Processes. The Sparse Variational Gaussian Processes (SVGP) framework enables efficient Gaussian Process inference by selecting inducing points. It achieves this by maximizing the Evidence Lower Bound (ELBO), which approximates the true posterior, providing a scalable approach to GP inference (Titsias, 2009).

Reproducing Kernel Hilbert Space (RKHS). Recent studies have highlighted the similarity between the RKHS of the Laplace kernel and NTK, shedding light on the NN-GP equivalence (Zheng & Li, 2023). This insight is crucial for understanding the theoretical underpinnings of NTK-based methods and their applications in dataset distillation.

Our work builds on these foundations, enhancing dataset distillation with NTKs and incorporating differential privacy guarantees to provide a robust, efficient, and privacy-preserving approach to data distillation.

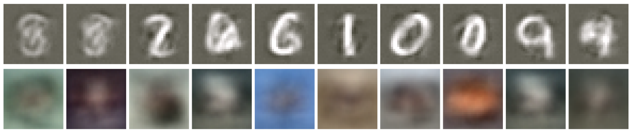

Figure 1. A distilled set using LATENT DISTILLATION; TOP: Distilled MNIST samples, obtained by passing inducing points through the decoder. BOTTOM: Similar, for CIFAR-10.

3. Our Approach

Algorithm 1 LATENT DISTILLATION: Our proposed method for dataset distillation proceeds by first constructing the FINITE NTK of the trained neural network, using the computed kernel for variational inducing point estimation for the surrogate Gaussian process. For small cases, initialization via a Determinantal Point Process sampling provides a warm-start for our inducing points.

```
Require: Data ($X_0$, $y_0$), NEURAL NETWORK $\Phi_\theta$, AUTOENCODER $A$: $A_d(A_e(X_0)) = X_0 + \epsilon$.
$X_1 \leftarrow A_e(X_0)$
$\theta_{\text{OPT}} \leftarrow$ MAXIMIZE$_\theta$ $\mathcal{L}(y_1 \mid X_1, \theta)$
DEFINE $k_{\text{NTK}}(x_i, x_j) = \mathbf{J}(x_i; \theta_{\text{OPT}})^T \mathbf{J}(x_j; \theta_{\text{OPT}})$
INITIALIZE $f_0 \sim \mathcal{GP}(0, k = k_{\text{NTK}}) = \mathcal{GP}(0, k = \mathbf{J}(x_i; \theta_{\text{OPT}})^T \mathbf{J}(x_j; \theta_{\text{OPT}}))$
$\mathcal{L}_0 \leftarrow \log \mathcal{L}(y_1)$, the marginal log likelihood of $f_0$ w.r.t $(X_1, y_1)$.
    We shall now proceed to define the variational optimization procedure.
INITIALIZE inducing points $\mathbf{Z}$ via DETERMINANTAL POINT PROCESS, $\mathcal{D}((X_1, y_1), k_{\text{NTK}})$
INITIALIZE $f_v = \mathcal{GP}(0, k = k_{\text{NTK}})$
DEFINE variational distribution $\mathcal{N}(\mathbf{u} \mid \mathbf{m}, \mathbf{S})$, where $\mathbf{u}$ are the latent function values of $\mathbf{Z}$.
$\mathbf{Z}_{\text{OPT}} \leftarrow$ MAXIMIZE$_{\mathbf{Z}}$ ELBO: $\mathcal{L}(\mathbf{m}, \mathbf{S}, \mathbf{Z}) \sim$ MINIMIZE$_{\mathbf{Z}}$ $(\mathcal{L}_0 - \text{ELBO}) = \mathcal{KL}[Q \mid\mid \hat{P}]$
    return $(A_d(\mathbf{Z}_{\text{OPT}}), \mathcal{GP}(\mathbf{Z}_{\text{OPT}}))$ as the CORESET of $\Phi$, for $(X_1, y_1)$
```

We train the downstream neural network on the soft targets over the inducing points returned by the Gaussian process trained in the latent space. This incorporates more information about each point, and results in enhanced accuracy, similar to techniques in Knowledge Distillation (Hinton et al., 2015).

4. Bounding the number of Inducing Points

Here, we give an intuition behind computing a bound on the number of inducing points, m (also, the size of the subset), for our proposed method.

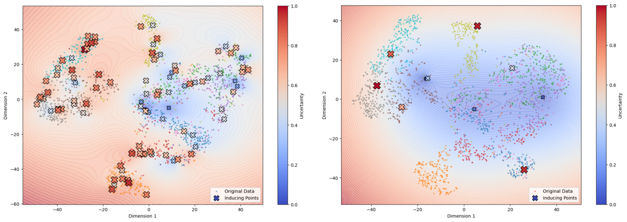

Figure 2. t-SNE plot of chosen inducing points for the Fashion MNIST dataset, for **100 inducing points** (roughly 10 img/cls), and **10 inducing points** (roughly 1 img/cls). We overlay uncertainty information from the inducing points, estimated through the variance of the variational distribution, to assess model confidence.

(Burt et al., 2020) showed that the KL-divergence for GPR, $KL(Q||P)$, can be made arbitrarily small with $m \ll n$ inducing points. It was shown that, for the Squared Exponential (SE) kernel, $m = O(\log(n)^d)$, is sufficient for performing inference. Under certain assumptions on the kernel and the distribution of the observed labels $\mathbf{y}$, if the initial inducing points are sampled according to an ϵ approximate m-DPP with K_{ff} as the kernel matrix, then $KL(Q||P)$ can be bounded as

$$\mathbb{E}[KL(P||Q)] \leq \frac{(m+1)n \sum_{r=m+1}^{\infty} \lambda_r + 2nv\epsilon}{\sigma^2} \tag{1}$$

where $K_{ff}[x, x] = k(x, x) < v, \forall x \in \mathbb{R}^d$ and λ_r is the r largest eigenvalue of K_{ff}. We aim to do develop similar bounds for the case of the NTK.

The convergence bound given in Equation (1) depends on the summation $\sum_{r=m+1}^{\infty} \lambda_r$ of the smallest eigenvalues of the kernel K. However, closed form expressions for this summation are known only for a handful of kernels such as the SE kernel.

(Chen et al., 2021) proved that when the datapoints are restricted to the d-dimensional hypersphere, $\mathbb{S}^{d-1}$, the reproducing kernel Hilbert space (RKHS) of the NTK of a deep neural network and the Laplace kernel, Equation (**??**), contain the same set of functions. Previously, (**?**) showed that the NTK for fully connected networks with ReLU activation, referred to as the ReLU-NTK for the remainder of this section, is similar to the Laplace kernel and that their eigenvalues decay polynomially at the same rate. Theorem 1 of the paper shows that the eigenvalues of the ReLU-NTK decays at a rate of $O(r^{-d})$, i.e, $\lambda_r \approx O(r^{-d})$. A similar result was shown in (**?**) for the case of a 2-layer ReLU-NTK.

Using these eigenvalue decay rates, Proposition 33 of (**?**), states that the number of inducing points is $O(n^\zeta)$, where

$\zeta \in (0, \frac{d-1}{d(4+d)})$. This suggests an upper bound on the number of inducing points as $O(n^{\frac{d-1}{d+4}})$. It remains an open question to derive tighter bounds for the size of the inducing points set. It should be noted that the bound in Equation (1) and hence, the bound on m holds when the initial set of inducing points is sampled using ϵ m-DPP. We now discuss this in more depth and present our primary results.

Lemma 1 (Eigenvalue Decay of Kernel Operators)**.** *Let k be a continuous kernel on $\mathbb{R}^D$, and μ be a measure on $\mathbb{R}^D$ with density p. The associated kernel integral operator $\mathcal{K}$ has eigenvalues bounded by:*

$$C_1 m^{-\eta} \leq \lambda_m \leq C_2 m^{-\eta} \tag{2}$$

for all $m \geq 1$, some $\eta > 1$ (i.e., polynomial decay), and arbitrary constants $C_1, C_2 > 0$.

Lemma 2 (NTK-Laplace Kernel Correspondence)**.** *The RKHS of the Neural Tangent Kernel is identical to that of the Laplace Kernel for data constrained to the hypersphere S^{d-1}, with the same eigenvalue decay of their associated kernel operator. Specifically, this implies that the NTK is a polynomial kernel for $d \geq 2$ with (Geifman et al., 2020):*

$$\eta_{NTK} = c_{NTK} d \tag{3}$$

where c_{NTK} is an arbitrary constant, empirically close to 1.

Theorem 1 (Optimal Inducing Point Size Estimation)**.** *Given the conditions in Lemmas 1 and 2:*

a) An inducing point subset of size $\mathcal{O}(N^{\zeta})$, where

$$\zeta \in \left(0, \frac{\eta - 1}{\eta(4+\eta)}\right) \tag{4}$$

is sufficient to derive a good approximation for the exact posterior Gaussian process with a variational approximation, such that:

$$KL[Q \| P] = \Omega(N^{1-\eta\zeta}) \tag{5}$$

where Q is the approximate Gaussian posterior and P is the exact posterior.

b) For the Neural Tangent Kernel, an inducing point subset of size $\mathcal{O}(N^{\zeta_{NTK}})$, where

$$\zeta_{NTK} \in \left(0, \frac{\eta_{NTK} - 1}{\eta_{NTK}(4 + \eta_{NTK})}\right) \tag{6}$$

is required to approximate the full Gaussian posterior mean.

c) The best approximation, resulting in the lowest value of $KL[Q\|P]$, can be achieved by choosing

$$|z| = \mathcal{O}(N^{\frac{d-1}{d+4}}) \text{ inducing points} \tag{7}$$

Proof. We begin with the result from Rasmussen and Wilk [2020] that Gaussian kernels with exponentially decaying operators can approximate the full posterior with a $\log(N)$ approximation of the full dataset of N samples.

For kernels with polynomial decay as described in Lemma 1, we can derive a bound on the size of the inducing point subset. Let $\mathcal{O}(N^{\zeta})$ be the size of this subset. The quality of the approximation, measured by $KL[Q\|P]$, is a decreasing function of $\eta\zeta$, bounded by:

$$\eta\zeta \leq \frac{\eta - 1}{\eta + 4} \tag{8}$$

This bound is itself increasing in η. Solving for ζ, we obtain the range given in the theorem.

Geifman and Yadav [2020] showed that the Reproducing Kernel Hilbert Spaces of the Neural Tangent Kernel is identical to that of the Laplace Kernel for data constrained to the hypersphere, with the same eigenvalue decay of their associated operators. Specifically, the eigenvalues of the Neural Tangent Kernel operator are bounded by:

$$C_1 m^{-d} \leq \lambda_m \leq C_2 m^{-d} \tag{9}$$

Combining this with Lemma 2, we can substitute $\eta_{NTK} = c_{NTK} d$ into our previous results. This gives us the bound for ζ_{NTK} as stated in part (b) of the theorem.

For the optimal approximation, we note that $KL[Q\|P]$ is a decreasing function of $(\eta\zeta)$, bounded by $\frac{\eta-1}{\eta+4}$. For the NTK, with $\eta_{NTK} = c_{NTK} d$ and c_{NTK} empirically close to 1, we can approximate this as:

$$\frac{\eta_{NTK} - 1}{\eta_{NTK} + 4} \approx \frac{d-1}{d+4} \tag{10}$$

Therefore, choosing $\mathcal{O}(N^{\frac{d-1}{d+4}})$ inducing points provides the best bounds on $KL[Q\|P]$. □

5. Convergence of Variational Approximation

Building upon our previous results on the bounds for NTK inducing points, we now demonstrate that these bounds ensure the convergence of the variational approximation to the true posterior. Specifically, we show that the KL divergence between the *approximate posterior* Q from the Sparse Gaussian process and the true posterior $\hat{P}$ from the Gaussian process over the full data approaches zero as the number of inducing points increases, and that empirically, $\mathcal{O}(N^{\frac{d-1}{d+4}})$ is enough to achieve a good approximation of the full data. We now leverage this result to show the convergence of $KL[Q\|\hat{P}]$ to zero.

We start with the decomposition of the KL divergence as derived in [Mathhews 2016 AISTATS]:

$$KL[Q\|\hat{P}] = KL[Q_Z\|P_Z] - \mathbb{E}_{Q_D}[\log L(Y|f_D)] + \log L(Y) \tag{11}$$

where,

1. $KL[Q_Z\|P_Z]$ is the KL divergence between the variational distribution over inducing points and the prior.
2. $\mathbb{E}_{Q_D}[\log L(Y|f_D)]$ is the expected log-likelihood under the variational predictive distribution.
3. $\log L(Y)$ is the log marginal likelihood of the full data.

Experimentally, $\log L(Y)$ can be calculated by computing the Exact Gaussian process over the full data, and the other two terms are computed as part of the variational approximation, during the calculation of the $ELBO$. For a good approximation of the full data via a set of inducing points z, $KL[Q\|\hat{P}]$ should become vanishingly small as the inducing points are optimized. [Figure] shows the training process and the change in $KL[Q\|\hat{P}]$ across iterations. We observe that as the inducing points are optimized, the downstream accuracy of the neural network trained on these points increases, and $KL[Q\|\hat{P}]$ decreases.

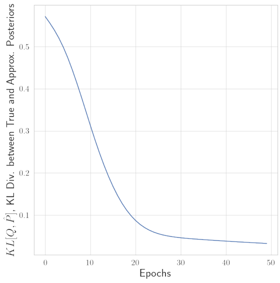

Figure 3. The KL Divergence $KL[Q,\hat{P}]$ between the APPROXIMATE POSTERIOR on the distilled data and the TRUE POSTERIOR based on the full data steadily decreases during the GP DISTILLATION process.

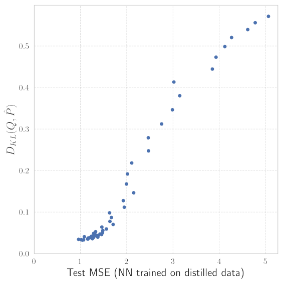

Figure 4. The decrease in KL Divergence expectedly corresponds to a steady decrease in the mean-squared error of the downstream Neural Network trained on the DISTILLED DATASET over the data from the same distribution.

Figure 5. KL Divergence and neural network performance during the distillation process.

6. Experimentation, Ablation, and Time Complexity

6.1. Dataset and Preprocessing

We conduct our primary experiments on the FashionMNIST, CIFAR-10 and MNIST datasets, which consists of 60,000 grayscale images (28 × 28) across 10 categories. The dataset is divided into 50,000 training and 10,000 test samples.

6.2. Model Architectures, Training and Optimization

Autoencoder: We use a fully connected autoencoder with a latent dimensionality of d_l (16, 32, or 64). The encoder comprises three linear layers: $784 \to 256 \to 128 \to d_l$, each followed by ReLU activations. The decoder mirrors this architecture: $d_l \to 128 \to 256 \to 784$, also with ReLU activations. The autoencoder is trained using the Adam optimizer with a learning rate of 1×10^{-3}. The model is trained for 750 epochs, with Mean Squared Error (MSE) as the reconstruction loss. We note that prolonged training of the autoencoder results in lower accuracy of the GPR in the latent space.

Gaussian Process Regression (GPR): We adopt a latent sparse Gaussian Process regression model based on GPyTorch. A variational strategy with 10, 50, and 100 inducing points is used, repeated across the latent dimensions. The GPR model is trained using a variational ELBO loss for 2000 epochs. We use the Adam optimizer with a learning rate of 3×10^{-2}, optimizing both the GP and likelihood parameters. Accuracy is computed by taking the maximum a posteriori probability from the predicted distribution.

Neural Network: For final classification, we use a fully connected neural network with two layers: $784 \to 128 \to 10$. ReLU activations are applied after the first layer. It is trained using the Adam optimizer with a learning rate of 1×10^{-3}, on the **soft labels** returned by the GPR. It is trained for 500 epochs, and accuracy is measured by comparing predicted labels with ground truth.

6.3. Evaluation and Ablations

We evaluate on a wide series of benchmarks, and conduct ablations. The projection into latent space via the autoencoder is necessary only for very high-dimensional datasets, so we first evaluate our method on simpler datasets from the UCI dataset repository. For faster optimization, we initialize the inducing points with a **determinantal point process** based sampling scheme.

To evaluate our theoretical results, we benchmark our methods on data generated via smooth functions on the d-dimensional hypersphere, with an appropriate classification threshold.

Time Complexity: Recent works [Loo et al.] have advocated for the use of Random Feature approximations (RFAD) for NNGP kernels for faster kernel matrix computation as compared to the empirical NTK as described in the original NTK [Nguyen et al.] paper, which reduces the complexity of the time-consuming kernel evaluation

Dataset	Algorithm	5 img/cl.	10 img/cl.
MNIST	GPD → NN	0.289	0.312
	LD → NN	**0.884** ($d_l : 32$)	0.902 ($d_l : 32$)
		0.865 ($d_l : 16$)	**0.916** ($d_l : 64$)
	KIP → NN	-	0.889
CIFAR-10	GPD → NN	0.101	0.094
	LD → NN	0.302 ($d_l : 32$)	0.326 ($d_l : 32$)
		-	0.332 ($d_l : 64$)
	KIP → NN	**-**	**0.362**
Fash.MNIST	LD → NN	0.302 ($d_l : 32$)	0.802 ($d_l : 32$)
		−	0.822 ($d_l : 64$)
	KIP → NN	**-**	**0.868**

Dataset	Algorithm	5 img/cl.	10 img/cl.
MNIST	LD → GP	0.901	0.937
	KIP → KRR	-	0.966
CIFAR-10	LD → GP	0.565	0.624
	KIP → KRR	-	0.405
Fash.MNIST	LD → GP	0.624	0.836
	KIP → KRR	-	0.876

Table 1. For MNIST and Fashion-MNIST, Latent Distillation is faster, and more accurate. The **left** table shows downstream performance when a neural network is trained on soft labels returned by the Gaussian process. The right table shows the raw accuracy of the Gaussian process on the inducing points, compared to accuracy on KRR by KERNEL INDUCING POINTS.
Note: **LD** stands for Latent Distillation and **GPD** stands for Gaussian Process Distillation, our proposed algorithms. Arrows indicate direction of transfer. **KIP** stands for Kernel Inducing Points [Ngyuen 2021 et al.]

Dataset	Algorithm	No. of Inducing Points		
		$m = 8$	$m = 16$	$m = 32$
Breast Cancer, $\|N\| = 569, d = 30$	GP DISTILLATION	**0.980**	**0.978**	**0.979**
	Coreset w/ DPP	0.900	0.892	0.943
	Uniform Random Coreset	0.833	0.906	0.953
		$m = 5$	$m = 10$	$m = 20$
Ionosphere, $\|N\| = 351, d = 34$	GP DISTILLATION	**0.894**	**0.920**	**0.911**
	Coreset w/ DPP	0.649	0.803	0.857
	Uniform Random Coreset	0.586	0.777	0.849
		$m = 5$	$m = 10$	$m = 30$
Heart Disease $\|N\| = 303, d = 13$	GP DISTILLATION	**0.874**	**0.862**	**0.885**
	Coreset w/ DPP	0.722	0.648	0.833
	Uniform Random Coreset	0.788	0.733	0.762

Table 2. GP DISTILLATION on a variety of Classification tasks, compared to a random sampling of points. While random samples are usually strong coresets, especially when the coreset itself is sizeable, our algorithm produces pseudo-datapoints that outperform the uniformly random case. The DPP itself is evidently not a strong coreset, but serves as a good initialization for the Gaussian process due to guarantees of diversity.

Coreset size, out of 200	90	45	15	6
Dimensions d	32	16	8	4
Random Initialization	0.735	0.665	0.650	0.620
DPP Initialization.	0.740	0.655	0.640	0.680
GPD w/ RBF Kernel	**0.975**	0.855	0.845	0.925
GPD w/ NTK	0.955	**0.915**	**0.905**	**0.950**
GPD w/ eNTK	0.775	0.770	0.850	0.710

Table 3. **Data on a Hypersphere** S^{D-1}**:** Comparing our method to other ways and different kernels for selecting a representative subset of the data. The ground-truth is a simple analytical function, thresholded to ensure equitable binary class distribution.

step from $\mathcal{O}(|T||S| + |S|^2)$ to $\mathcal{O}(|T| + |S|)$, resulting in roughly two orders of magnitude in speedups, empirically. Since our method also involves these steps, we can expect similar performance increases on our algorithm, *in addition* to the speedups offered by optimizing in the latent space. Our basic algorithm itself results in roughly 3-4x speedups over RFAD for similar downstream performance. For example, on Fashion-MNIST, we see 86.8% accuracy with LATENT DISTILLATION in 46.11 seconds (1.90 seconds for the autoencoder training, and 44.21 seconds for the GPR in latent space to reach 90% accuracy). RFAD requires 459.5 seconds to cross 85% accuracy – however, with prolonged training, RFAD surpasses our algorithm in accuracy at around 10 minutes of training.

Choice of Kernels: For simple datasets, we show that the infinite-width NTK outperforms other kernels when it comes to downstream accuracy. For larger datasets, the ease of optimization of the RBF kernel wins out over the benefits of choosing the NTK (both in the infinite-width and empirical case).

References

1. Burt, D. R., Rasmussen, C. E., van der Wilk, M. (2020). Convergence of Sparse Variational Inference in Gaussian Processes Regression. *Journal of Machine Learning Research*, 21(131), 1-63. Retrieved from http://jmlr.org/papers/v21/19-1015.html

2. Geifman, A., Yadav, A., Kasten, Y., Galun, M., Jacobs, D., Basri, R. (2020). On the similarity between the Laplace and Neural Tangent Kernels. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Article No. 123). Vancouver, BC, Canada: Curran Associates Inc.

3. Titsias, M. (2009). Variational Learning of Inducing Variables in Sparse Gaussian Processes. In D. van Dyk M. Welling (Eds.), *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics* (Vol. 5, pp. 567-574). Hilton Clearwater Beach Resort, Clearwater Beach, Florida, USA: PMLR.

4. Burt, D. R., Rasmussen, C. E., van der Wilk, M. (2019). Rates of Convergence for Sparse Variational Gaussian Process Regression. In K. Chaudhuri R. Salakhutdinov (Eds.), *Proceedings of the 36th International Conference on Machine Learning* (Vol. 97, pp. 862-871). PMLR.

5. Hensman, J., Fusi, N., Lawrence, N. D. (2013). Gaussian processes for Big Data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence* (pp. 282-290). Bellevue, WA: AUAI Press.

6. Matthews, A. G. de G., Hensman, J., Turner, R. E., Ghahramani, Z. (2015). On Sparse Variational Methods and the Kullback-Leibler Divergence between Stochastic Processes. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*.

7. Chen, L., Xu, S. (2021). Deep Neural Tangent Kernel and Laplace Kernel Have the Same RKHS. In *International Conference on Learning Representations*. Retrieved from https://openreview.net/forum?id=vK9WrZ0QYQ

8. Bietti, A., Mairal, J. (2019). On the inductive bias of neural tangent kernels. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (Article No. 1155). Red Hook, NY: Curran Associates Inc.

9. Neal, R. M. (1996). Priors for Infinite Networks. In *Bayesian Learning for Neural Networks* (pp. 29-53). New York, NY: Springer New York. https://doi.org/10.1007/978-1-4612-0745-0_2

10. Williams, C. (1996). Computing with Infinite Networks. In M. C. Mozer, M. Jordan, T. Petsche (Eds.), *Advances in Neural Information Processing Systems* (Vol. 9). MIT Press.

11. Hazan, T., Jaakkola, T. (2015). Steps Toward Deep Kernel Methods from Infinite Neural Networks. Retrieved from https://arxiv.org/abs/1508.05133

12. Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., Sohl-Dickstein, J. (2017). Deep Neural Networks as Gaussian Processes. Retrieved from https://arxiv.org/abs/1711.00165

13. Matthews, A. G. de G., Hron, J., Rowland, M., Turner, R. E., Ghahramani, Z. (2018). Gaussian Process Behaviour in Wide Deep Neural Networks. In *International Conference on Learning Representations*. Retrieved from https://openreview.net/forum?id=H1-nGgWC-

14. Novak, R., Xiao, L., Bahri, Y., Lee, J., Yang, G., Abolafia, D. A., Pennington, J., Sohl-Dickstein, J. (2019). Bayesian Deep Convolutional Networks with Many Channels are Gaussian Processes. In *International Conference on Learning Representations*. Retrieved from https://openreview.net/forum?id=B1g30j0qF7

15. Yang, G. (2021). Tensor Programs I: Wide Feedforward or Recurrent Neural Networks of Any Architecture are Gaussian Processes. Retrieved from https://arxiv.org/abs/1910.12478

16. Jacot, A., Gabriel, F., Hongler, C. (2018). Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (pp. 8580-8589). Red Hook, NY: Curran Associates Inc.

17. Lee, J., Xiao, L., Schoenholz, S. S., Bahri, Y., Novak, R., Sohl-Dickstein, J., Pennington, J. (2020). Wide neural networks of any depth evolve as linear models under gradient descent. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12), 124002. https://doi.org/10.1088/1742-5468/abc62b

18. Golikov, E., Pokonechnyy, E., Korviakov, V. (2022). Neural Tangent Kernel: A Survey. Retrieved from https://arxiv.org/abs/2208.13614

19. Novak, R., Sohl-Dickstein, J., Schoenholz, S. S. (2022). Fast Finite Width Neural Tangent Kernel. Retrieved from https://arxiv.org/abs/2206.08720

20. Mohamadi, M. A., Bae, W., Sutherland, D. J. (2023). A fast, well-founded approximation to the empirical neural tangent kernel. In *Proceedings of the 40th International Conference on Machine Learning*. Honolulu, Hawaii, USA: JMLR.org.

21. Drineas, P., Mahoney, M. W., Muthukrishnan, S. (2006). Sampling algorithms for l2 regression and applications. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm* (pp. 1127-1136). Miami, Florida: Society for Industrial and Applied Mathematics.

22. Coleman, C., Yeh, C., Mussmann, S., Mirzasoleiman, B., Bailis, P., Liang, P., Leskovec, J., Zaharia, M. (2020). Selection via Proxy: Efficient Data Selection for Deep Learning. Retrieved from https://arxiv.org/abs/1906.11829

23. Guo, C., Zhao, B., Bai, Y. (2022). DeepCore: A Comprehensive Library for Coreset Selection in Deep Learning. In *Database and Expert Systems Applications: 33rd International Conference, DEXA 2022, Vienna, Austria, August 22–24, 2022, Proceedings, Part I* (pp. 181-195). Berlin, Heidelberg: Springer-Verlag. https://doi.org/10.1007/978-3-031-12423-5_14

24. Kaushal, V., Iyer, R., Kothawade, S., Mahadev, R., Doctor, K., Ramakrishnan, G. (2019). Learning From Less Data: A Unified Data Subset Selection and Active Learning Framework for Computer Vision. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 1289-1299). IEEE. https://doi.org/10.1109/WACV.2019.00142

25. Killamsetty, K., Sivasubramanian, D., Ramakrishnan, G., De, A., Iyer, R. (2021). GRAD-MATCH: Gradient Matching based Data Subset Selection for Efficient Deep Model Training. Retrieved from https://arxiv.org/abs/2103.00123

26. Killamsetty, K., Sivasubramanian, D., Ramakrishnan, G., Iyer, R. (2021). GLISTER: Generalization based Data Subset Selection for Efficient and Robust Learning. Retrieved from https://arxiv.org/abs/2012.10630

27. Killamsetty, K., Zhao, X., Chen, F., Iyer, R. K. (2021). RETRIEVE: Coreset Selection for Efficient and Robust Semi-Supervised Learning. Retrieved from https://arxiv.org/abs/2106.07760

28. Mindermann, S., Brauner, J. M., Razzak, M. T., Sharma, M., Kirsch, A., Xu, W., Höltgen, B., Gomez, A. N., Morisot, A., Farquhar, S., Gal, Y. (2022). Prioritized Training on Points that are Learnable, Worth Learning, and not yet Learnt. In K. Chaudhuri et al. (Eds.), *Proceedings of the 39th International Conference on Machine Learning* (Vol. 162, pp. 15630-15649). PMLR.

29. Mirzasoleiman, B., Bilmes, J., Leskovec, J. (2020). Coresets for Data-efficient Training of Machine Learning Models. Retrieved from https://arxiv.org/abs/1906.01827

30. Durga, S., Iyer, R., Ramakrishnan, G., De, A. (2021). Training Data Subset Selection for Regression with Controlled Generalization Error. In M. Meila T. Zhang (Eds.), *Proceedings of the 38th International Conference on Machine Learning* (Vol. 139, pp. 9202-9212). PMLR.

31. Jain, E., Nandy, T., Aggarwal, G., Tendulkar, A., Iyer, R., De, A. (2024). Efficient data subset selection to generalize training across models: transductive and inductive networks. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*. New Orleans, LA, USA: Curran Associates Inc.