

---

# Loss-Curvature Matching for Dataset Selection and Condensation

---

Seungjae Shin\*  
KAIST

Heesun Bae\*  
KAIST

Donghyeok Shin  
KAIST

Weonyoung Joo  
Ewha Womans University

Il-Chul Moon†  
KAIST

## Abstract

Training neural networks on a large dataset requires substantial computational costs. Dataset reduction selects or synthesizes data instances based on the large dataset, while minimizing the degradation in generalization performance from the full dataset. Existing methods utilize the neural network during the dataset reduction procedure, so the model parameter becomes important factor in preserving the performance after reduction. By depending upon the importance of parameters, this paper introduces a new reduction objective, coined LCMat, which Matches the Loss Curvatures of the original dataset and reduced dataset over the model parameter space, more than the parameter point. This new objective induces a better adaptation of the reduced dataset on the perturbed parameter region than the exact point matching. Particularly, we identify the worst case of the loss curvature gap from the local parameter region, and we derive the implementable upper bound of such worst-case with theoretical analyses. Our experiments on both coreset selection and condensation benchmarks illustrate that LCMat shows better generalization performances than existing baselines.

## 1 INTRODUCTION

Although we live in the world of big data, utilizing such big data induces a considerable amount of time and space complexity in the learning process (Mirzasoaleiman et al., 2020; Kim et al., 2022a; Patterson et al., 2021). Accordingly, researchers introduced a concept of *dataset selection* and *dataset condensation*, etc (Killamsetty et al., 2021b; Paul et al., 2021). These concepts state that a dataset with smaller cardinality may yield similar performance in machine learning compared to a big dataset, if the smaller dataset delivers

all task-relevant information as the original dataset. Dataset reduction provides tangible benefits because the reduced dataset will consume less time in training and less space in memory (Pooladzandi et al., 2022). Moreover, such benefits are the desiderata of some well-known tasks, i.e. continual learning with memory replay (Lopez-Paz and Ranzato, 2017; Borsos et al., 2020).

As we reviewed, there exist two approaches in reducing the cardinality of dataset: the selection-based method (a.k.a. dataset selection) and the condensation-based method (a.k.a. dataset condensation). While these are similar concepts in terms of reducing data cardinality without performance degradation, both approaches have been treated and researched in different papers. Hence, this paper will refer to these approaches by a unifying term of *dataset reduction*. 1) Selection-based method optimally selects a small set of data instances out of the full dataset with an expectation on the identical task-relevant information of the small and the full datasets (Agarwal et al., 2020; Sener and Savarese, 2018; Welling, 2009). In contrast, 2) condensation-based method synthesizes the data instances by directly passing the learning gradient to the data input (Zhao and Bilén, 2021b; Nguyen et al., 2021).

To identify the examples which contribute the most to learning, both lines of work mainly utilize the gradient matching between the original dataset and reduced dataset (Mirzasoaleiman et al., 2020; Killamsetty et al., 2021a; Zhao et al., 2020), which provides theoretical analyses unlike other methods (Coleman et al., 2019; Zhao and Bilén, 2021b). However, gradient matching is conducted at a specific model parameter, so this implementation would fundamentally be biased by the model parameter at hand. Therefore, the generalization over the perturbed parameter point could be potentially beneficial. From the perspective of generalization over the model parameter region, the gradient matching can be generally extended to the local curvature matching in the response surface. Recently, Sharpness-Aware Minimization (SAM) (Foret et al., 2020) has made breakthroughs which ensure the generalization of the model by regularizing the flat minima over the local parameter region, not the point estimate of the parameter. This opens a new possibility of applying the spirit of SAM to the dataset reduction field.

This paper introduces a new objective for dataset reduction,

---

\* Equal contribution. † Corresponding author.  
Proceedings of the 26<sup>th</sup> International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206. Copyright 2023 by the author(s).

coined **Loss-Curvature Matching** (LCMat), which matches the loss curvature of the original dataset and the resulting reduced dataset on the target parameter region. This matching could be also interpreted as the sharpness of the loss difference between two datasets. This notion enables LCMat as the first work of sharpness-aware dataset reduction. This merge of dataset reduction and sharpness-aware minimization induces two contributions. First, SAM only provides the optimization based on the model parameter, whereas the optimization of dataset reduction is conducted based on the input data variable. To enable the input-based optimization on the defined sharpness, this paper derives an implementable upper bound of the sharpness, which results in an objective of LCMat. Second, we adaptively transform the objective into the function of either selection or condensation objective, so LCMat becomes the fundamentally applicable mechanism for dataset reduction overarching the dataset selection as well as the dataset condensation. We conduct experiments over the evaluation scenarios with different benchmark datasets, and we confirm that LCMat shows clear merit when the reduction ratio becomes significant and when the evaluation scenario becomes dynamic and complex, e.g. continual learning.

## 2 PRELIMINARY

### 2.1 Notations

This paper focuses on dataset reduction for classification tasks, which is a widely studied scenario in the community of dataset reduction (Mirzasoleiman et al., 2020; Welling, 2009; Zhao et al., 2020). Assuming a classification into  $c$  classes, let  $\mathcal{X} \in \mathbb{R}^d$  and  $\mathcal{Y} = \{1, 2, \dots, c\}$  be input variable space and a label candidate set, respectively. Given  $\mathcal{X}$  and  $\mathcal{Y}$ , our training dataset is  $T = \{(x_i, y_i)\}_{i=1}^n \subseteq \mathcal{X} \times \mathcal{Y}$ . We assume that each training instance  $(x, y)$  is drawn i.i.d from the population distribution  $\mathbb{D}$ .

Let a classifier  $f_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^c$  be parameterized by  $\theta \in \Theta$ . Under this definition, the training loss on  $T$  and the population loss on  $\mathbb{D}$  are denoted as  $\mathcal{L}(T; \theta) = \frac{1}{n} \sum_{i=1}^n \ell(x_i, y_i; \theta)$  and  $\mathcal{L}(\mathbb{D}; \theta) = \mathbb{E}_{(x,y) \sim \mathbb{D}}[\ell(x, y; \theta)]$ , respectively. Here,  $\ell$  means a value of loss function for a pair of  $x$  and  $y$ <sup>1</sup>.

### 2.2 Previous Researches on Dataset Reduction

This paper focuses on *dataset reduction*, whose purpose is to generate a cardinality-reduced dataset  $S$  from the training dataset  $T$ , as such  $|S| \ll |T|$ , while maximally preserving the task-relevant information from  $T$ .

**Selection-based Methods** Selection-based methods (Welling, 2009; Sener and Savarese, 2018) find a data subset  $S \subset T$  that satisfies the cardinality constraint while

maximizing the objective defined by the informativeness of  $S$ . The approximated objectives are defined by utilizing either 1) gradient (Paul et al., 2021; Mirzasoleiman et al., 2020; Killamsetty et al., 2021a), 2) loss (Toneva et al., 2018), 3) uncertainty (Coleman et al., 2019), and 4) decision boundary (Ducoffe and Precioso, 2018; Margatina et al., 2021). This section surveys existing methods with emphasis on gradient-based objectives because our method is primarily relevant to them. Gradient-based methods minimize the distance between the gradients from the training dataset  $T$ ; and the (weighted) gradients from  $S$  as follows:

$$\min_{\mathbf{w}, S} \mathcal{D} \left( \sum_{(x,y) \in T} \frac{\nabla_{\theta} \ell(x, y; \theta)}{|T|}, \sum_{(x,y) \in S} \frac{w_x \nabla_{\theta} \ell(x, y; \theta)}{\|\mathbf{w}\|_1} \right) \quad (1)$$

s.t.  $S \subset T, w_x \geq 0$

Here,  $\mathbf{w}$  is the vector of learnable weights for the data instances in subset  $S$ ;  $\|\mathbf{w}\|_1$  is l1 norm of  $\mathbf{w}$ ; and  $\mathcal{D}$  measures the distance between two gradients.

To solve the selection problem, Mirzasoleiman et al. (2020) converts Eq (1) into the submodular maximization problem, and this research utilizes the greedy approach to optimize Eq (1). Compared to Mirzasoleiman et al. (2020), Killamsetty et al. (2021a) utilizes orthogonal matching pursuit algorithm (Elenberg et al., 2018a) and  $L_2$  regularization term over  $\mathbf{w}$  to stabilize the optimization. Pooladzandi et al. (2022) replaces  $\nabla_{\theta} \ell(x, y; \theta)$  in Eq (1) with a preconditioned gradient with the Hessian matrix, which leverages the second-order information for optimization. Having said that, the optimization of Eq (1) is highly dependent on the given  $\theta$ , so the gradient matching could be potentially biased by the single snapshot  $\theta$  because the small-sized  $S$  would be vulnerable to selection bias to summarize  $T$ .

**Condensation-based Methods** Instead of selecting  $S$  from  $T$ , a small dataset,  $S$  can be directly synthesized to achieve the similar performance from  $T$  Wang et al. (2018). Then,  $S$  becomes a learnable variable updated via  $S \leftarrow S - \gamma \nabla_S \mathcal{L}(T, S)$ , where  $\mathcal{L}(T, S)$  is a general loss function which is dependent on both  $T$  and  $S$ . Zhao et al. (2020) proposed Dataset Condensation (DC), which matches the gradients between  $T$  and  $S$  over the optimization path of  $S$  as follows:

$$\min_S \mathbb{E}_{\theta^0 \sim P_{\theta^0}} \left[ \sum_k \mathcal{D}(\nabla_{\theta_S^k} \mathcal{L}(T; \theta_S^k), \nabla_{\theta_S^k} \mathcal{L}(S; \theta_S^k)) \right] \quad (2)$$

Here,  $\theta^0$  is the initialized parameter from  $P_{\theta^0}$ ; and  $\theta_S^k$  is the parameter updated with  $k$  iterations on SGD with  $S$ . The optimization of Eq (2) can be highly-dependent on the learning trajectory of  $\theta$  from  $S$ . Other condensation methods<sup>2</sup> utilize either 1) feature vectors (Zhao and Bilén,

<sup>1</sup>This paper utilizes cross-entropy as a loss function.

<sup>2</sup>See Appendix C.2 for detailed surveys

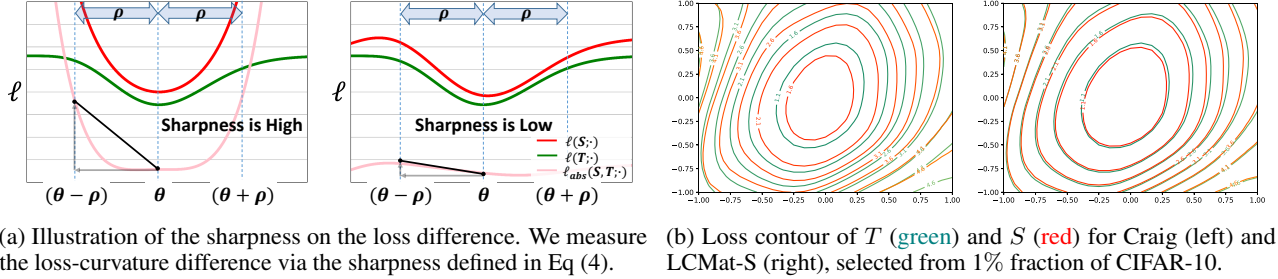


Figure 1: (a) The sharpness on loss differences represents the degree of difference on loss surfaces. (b) (left) The data subset,  $S$ , which is selected by Craig (Mirzasoleiman et al., 2020), does not match the loss curvature of the training dataset. (right) On the other hand, LCMat-S successfully matches the loss curvatures of  $T$  and  $S$ . We visualize the loss landscape according to the implementation of Li et al. (2018).

2021b; Wang et al., 2022) or 2) kernel products to propagate the task-relevant information of  $T$  into  $S$  (Nguyen et al., 2021). However, these methods do not provide theoretical analyses of the relation between  $T$  and  $S$ .

### 2.3 Generalization on Parameter Space

Apart from dataset reduction, a new research area has emerged by considering generalization over parameter space and its optimization (Sun et al., 2021; Wu et al., 2020; He et al., 2019). Several studies have focused on the problem of  $\theta$  over-fitting to  $T$  (Izmailov et al., 2018; Foret et al., 2020; Kim et al., 2022b), and they confirmed that optimization on the perturbed parameter region has a strong correlation to the generalization performance of the model. Sharpness-Aware Minimization (SAM) (Foret et al., 2020) is an optimizer for the model parameter, which regularizes the locality region of  $\theta$  to be the flat minima on the loss curvature as follows:

$$\min_{\theta} \max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}(T; \theta + \epsilon) \quad (3)$$

Here,  $\epsilon$  is the perturbation vector to the parameter; and  $\rho$  denotes the maximum size of the perturbation vector. As the objective is a function defined by both input and model parameter, it is possible to solve the generalization of a model parameter through the optimization of input data. However, there is no such study, which improves the generalization of the perturbed parameter space via optimizing the input data variable, to the best of our knowledge. It should be noted that adversarial training (Zhang et al., 2019) is different from our method because the perturbation for the worst case is conducted on the input space, not on the parameter space.

## 3 METHOD

As described in Section 2.2, recent methods in dataset reduction propagate the task-relevant information from  $T$  to  $S$  by aligning the gradients of a specific  $\theta$ . Given that dataset reduction hinges upon the utilization of  $\theta$ , the performance

depends on the trained  $\theta$  at the moment of reduction. Therefore, the optimal dataset reduction  $S^*$  would be different from  $S$ , which is biased by  $\theta$  at the specific state of  $f_{\theta}$ . Therefore, our research question becomes how to design a parameter-robust algorithm for dataset reduction while the algorithm still uses  $\theta$  by the necessity of the implementation practice.

### 3.1 Parameter Generalization in Dataset Reduction

A loss function  $\mathcal{L}$  quantifies the fitness of  $\theta$  under a certain dataset. Accordingly, the optimization of  $S$  toward  $T$  with respect to  $\theta$  would decrease  $|\mathcal{L}(T; \theta) - \mathcal{L}(S; \theta)|$ , which is the loss difference between  $T$  and  $S$  on  $\theta$ . However, if  $|\mathcal{L}(T; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)|$  increases with small perturbation  $\epsilon$  on  $\theta$ , then this increment indicates the lack of generalization on  $\theta + \epsilon$ , or an over-fitted reduction of  $S$  by  $\theta$ . This generalization failure on the locality of  $\theta$  subsequently results in the large difference of loss surfaces between  $T$  and  $S$ , as illustrated in Figure 1a. Figure 1a shows that the difference of loss surfaces between  $T$  and  $S$  could be measured by the sharpness of the loss differences, whose color is pink, on the target parameter region.

*Remark 3.1.* Assuming the strict convexity of  $\mathcal{L}$  over  $\Theta$ , if  $|\mathcal{L}(T; \theta) - \mathcal{L}(S; \theta)| = c$  for some fixed constant  $c \geq 0$  and any  $\theta \in \Theta$ ,  $\operatorname{argmin}_{\theta} \mathcal{L}(T; \theta) = \operatorname{argmin}_{\theta} \mathcal{L}(S; \theta)$ .

Remark 3.1 explains that the optimal  $\theta$  for  $T$  and  $S$  are the same if the loss difference is constant over the parameter space, which is the state when the loss curvatures of  $T$  and  $S$  are the same. If this condition is satisfied, we could safely utilize  $S$  for learning  $\theta$  where the generalization performance of  $\theta$  from  $S$  is guaranteed to be the same as that of  $T$ . This motivates us to match the loss curvatures between  $T$  and  $S$ , whose objective is introduced in the next section.

### 3.2 Loss-Curvature Matching (LCMat)

This section introduces a parameter-robust objective for dataset reduction, coined Loss-Curvature Matching

(LCMat), which matches the loss curvature of  $T$  and  $S$  based on a currently presented  $\theta$ . The target region of the objective is specified by the  $\rho$ -ball perturbed region of  $\theta$ . In Eq (3), SAM optimizes the worst-case sharpness from the target region of  $\theta$ , where the worst-case optimization becomes efficient when the optimization is requested over the specific region (Sagawa et al., 2019; Foret et al., 2020). Following the worst-case optimization scheme, we formulate the primary objective as follows:

$$\min_S \max_{\|\epsilon\|_2 \leq \rho} \frac{\mathcal{L}_{abs}(T, S; \theta + \epsilon) - \mathcal{L}_{abs}(T, S; \theta)}{\rho} \quad (4)$$

Here, we denote the loss difference between  $T$  and  $S$  on  $\theta$ ,  $\mathcal{L}_{abs}(T, S; \theta) = |\mathcal{L}(T; \theta) - \mathcal{L}(S; \theta)|$ . In Eq (4),  $S$  is optimized to minimize the sharpness of  $\mathcal{L}_{abs}(T, S; \theta)$  over the  $\rho$ -ball perturbed region from  $\theta$ . The optimization on Eq (4) incurs the maximization of  $\mathcal{L}_{abs}(T, S; \theta)$ , which could result in the overly under-fitted state of  $S$  on  $\theta$ . In our implementation,  $\mathcal{L}_{abs}(T, S; \theta)$  is bounded or regularized during the optimization. See Appendix B.1 for detailed analyses. Also, Eq (4) is defined on the case of single  $\theta$  for simplicity, and it could be generalized to any  $\theta \in \Theta$ .

The next question is how to optimize  $S$  by Eq (4). As our learning target is  $S$ , not  $\theta$ ; it is intractable to utilize SAM because SAM only provides the gradient of  $\theta$  for the corresponding sharpness. We introduce Proposition 3.2, which provides a tractable and differentiable upper bound of Eq (4) as follows:

**Proposition 3.2.** *When  $\mathbb{H}_D = \nabla_{\theta}^2 \mathcal{L}(D; \theta)$  is a Hessian matrix of  $\mathcal{L}(D; \theta)$ , let  $\mathbb{H}_{T,S} = \mathbb{H}_T - \mathbb{H}_S = \nabla_{\theta}^2 \mathcal{L}(T; \theta) - \nabla_{\theta}^2 \mathcal{L}(S; \theta)$  and  $\lambda_1^{T,S}$  be the maximum eigenvalue of the matrix  $\mathbb{H}_{T,S}$ , then we have: (Proof in Appendix A.1)*

$$\begin{aligned} & \max_{\|\epsilon\|_2 \leq \rho} \frac{\mathcal{L}_{abs}(T, S; \theta + \epsilon) - \mathcal{L}_{abs}(T, S; \theta)}{\rho} \\ & \leq \underbrace{\left\| \nabla_{\theta} \mathcal{L}(T; \theta) - \nabla_{\theta} \mathcal{L}(S; \theta) \right\|_2}_{\text{Gradient Matching via } L_2\text{-norm.}} \\ & \quad + \underbrace{\frac{1}{2} \rho \lambda_1^{T,S}}_{\text{Max eigenvalue}} + \max_{\|v\|_2 \leq 1} O(\rho^2 v^3) \end{aligned} \quad (5)$$

According to Proposition 3.2, the upper bound of the Eq (4) consists of 1) the  $L_2$  norm of gradient differences between  $T$  and  $S$ ; 2) the maximum eigenvalue of  $\mathbb{H}_{T,S}$ ; and 3) remaining higher-order terms. Given a certain selection of  $\rho$  determining the locality scope of the  $\theta$ , Proposition 3.2 argues that the gradient matching objective would not be enough for the loss surface matching if  $\lambda_1^{T,S}$  holds a large proportion in the upper bound.

Figure 2 shows the value of  $\|\nabla_{\theta} \mathcal{L}(T; \theta) - \nabla_{\theta} \mathcal{L}(S; \theta)\|_2$  and  $\frac{1}{2} \rho \lambda_1^{T,S}$  measured from different methods with  $\rho = 0.5$ . For the gradient matching term, all methods show similar

values, which means that these methods could not be distinguished by the learning from the gradient matching term. On the contrary,  $\lambda_1^{T,S}$  holds a large proportion and takes high variance across the tested methods, so the upper bound differences among the methods eventually rely on the value of  $\lambda_1^{T,S}$ . By excluding higher-order terms in Proposition 3.2, the resulting alternative objective is as follows:

$$\min_S \left\| \nabla_{\theta} \mathcal{L}(T; \theta) - \nabla_{\theta} \mathcal{L}(S; \theta) \right\|_2 + \frac{1}{2} \rho \lambda_1^{T,S} \quad (6)$$

Directly solving the optimization of Eq (6) requires an explicit calculation of the Hessian matrices,  $\mathbb{H}_T$  and  $\mathbb{H}_S$ . This calculation is too costly for over-parameterized models, such as neural networks. To overcome the computational overhead, various methods in machine learning have utilized the diagonal approximation of Hessian (Rame et al., 2022; Yao et al., 2021) as a common technique. According to Rame et al. (2022),  $\mathbb{H}$  becomes diagonally dominant at the end of training. We apply the diagonal approximation on  $\mathbb{H}_T$  and  $\mathbb{H}_S$ , and we denote the corresponding diagonal Hessian as  $\hat{\mathbb{H}}_T = \text{diag}(\mathbb{H}_T)$  and  $\hat{\mathbb{H}}_S = \text{diag}(\mathbb{H}_S)$ . When we replace  $\mathbb{H}_T$  and  $\mathbb{H}_S$  into  $\hat{\mathbb{H}}_T$  and  $\hat{\mathbb{H}}_S$ , respectively, Eq (6) is derived<sup>3</sup> as follows:

$$\min_S \left\| \nabla_{\theta} \mathcal{L}(T; \theta) - \nabla_{\theta} \mathcal{L}(S; \theta) \right\|_2 + \frac{1}{2} \rho \max_k \left| \hat{\lambda}_k^T - \hat{\lambda}_k^S \right| \quad (7)$$

Here,  $\hat{\lambda}_k^T$  and  $\hat{\lambda}_k^S$  are eigenvalues of  $\hat{\mathbb{H}}_T$  and  $\hat{\mathbb{H}}_S$  on  $k$ -th dimension for  $\theta$ . Having said that, we provide an adaptive application of our objective, Eq (7), on two approaches: selection-based methods and condensation-based methods, in Sections 3.3 and 3.4, respectively.

### 3.3 LCMat for Selection-based method

To select  $S \subseteq T$ , which minimizes Eq (7); we transform (7) into the selection-based objective with the cardinality constraint on  $S$ , in a sample-wise derivation as follows:

$$\begin{aligned} & \min_{S \subseteq T} \left( \left\| \frac{1}{|T|} \sum_{(x_i, y_i) \in T} \mathbf{g}_i^T - \frac{1}{|S|} \sum_{(x_j, y_j) \in S} \gamma_j \mathbf{g}_j^S \right\|_2 \right. \\ & \quad \left. + \frac{1}{2} \rho \max_k \left| \frac{1}{|T|} \sum_{(x_i, y_i) \in T} \hat{\lambda}_{i,k}^T - \frac{1}{|S|} \sum_{(x_j, y_j) \in S} \gamma_j \hat{\lambda}_{j,k}^S \right| \right) \\ & \text{s.t. } |S| \ll |T| \end{aligned} \quad (8)$$

<sup>3</sup>See Appendix A.2 for the proof.

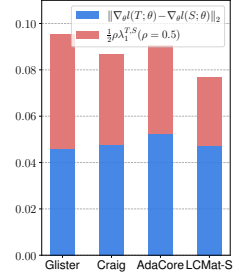


Figure 2: Report on each term in Eq (6) for the selected methods.  $\lambda_1^{T,S}$  holds a significant proportion.

Here, we denote the per-sample gradient as  $\mathbf{g}_i^T = \nabla_{\theta} \ell(x_i, y_i; \theta)$  for  $(x_i, y_i) \in T$ , and we also denote the  $k$ -th dimension eigenvalue of the per-sample Hessian as  $\hat{\lambda}_{i,k}^T$  for  $(x_i, y_i) \in T$ . Also, we introduce the learnable weight  $\gamma_j$  for  $(x_j, y_j) \in S$  to build Eq (8) as a generalized form.

It is well known the subset selection problem is NP-hard (Sener and Savarese, 2018; Mirzasoleiman et al., 2020). When we maximize  $\left| \frac{1}{|T|} \sum_{(x_i, y_i) \in T} \hat{\lambda}_{i,k}^T - \frac{1}{|S|} \sum_{(x_j, y_j) \in S} \gamma_j \hat{\lambda}_{j,k}^S \right|$  with respect to  $k$ ,  $k$  will be different by each subset  $S \subseteq T$ , where the search for  $k$  based on every possible  $S \subseteq T$  would be very costly. To relax the computational constraints on a search for  $k$ , we empirically optimize the following equation, which does not need the search of  $k$ , on behalf of the second term in Eq (8):

$$\frac{1}{2} \rho \sum_{k \in \mathcal{K}} \left| \frac{1}{|T|} \sum_{(x_i, y_i) \in T} \hat{\lambda}_{i,k}^T - \frac{1}{|S|} \sum_{(x_j, y_j) \in S} \gamma_j \hat{\lambda}_{j,k}^S \right| \quad (9)$$

Here,  $\mathcal{K}$  is a set of indexes for  $K$  sub-dimensions on  $\theta$ . We select  $K$  dominant sub-dimensions based on the variance of  $\hat{\lambda}_k^T = [\hat{\lambda}_{i,k}^T]_{i=1}^{|T|}$  for each  $k$ , which is denoted by the set  $\mathcal{K} = \operatorname{argmax}_{\mathcal{K}, |\mathcal{K}|=K} \sum_{j \in \mathcal{K}} \operatorname{Var}(\hat{\lambda}_k^T)$ . We empirically show that the true  $k$  in Eq (8) is always in  $\mathcal{K}$ , where the hyper-parameter of sub-dimensions  $K$  is fixed to 100 in our experiments. See Appendix B.2 for detailed analyses.

By the notion of regarding the subset selection as sparse vector approximation (Elenberg et al., 2018b; Mirzasoleiman et al., 2020), existing methods utilize submodular optimization with a simple greedy algorithm to get a nearly-optimal solution on their objectives. Similar to Mirzasoleiman et al. (2020), we utilize a facility location function (Lin et al., 2009; Lin and Bilmes, 2012) for the submodular optimization. The facility location function quantifies the cover of  $T$  given its subset  $S$  by summation of the similarities defined between every  $i \in T$  and its closest element  $j \in S$ . Formally, a facility location is defined as  $F(S) = \sum_{i \in T} \max_{j \in S} s_{i,j}$ , where  $s_{i,j}$  is the similarity between  $i, j \in T$ . By utilizing the analytical result of Craig, we get an upper bound of the error for Eq (9) as follows: (Proof in Appendix A.3)

$$\begin{aligned} \min_{S \subseteq T} \left\| \bar{\mathbf{g}}^T - \gamma^S \bar{\mathbf{g}}^S \right\|_2 + \frac{1}{2} \rho \sum_{k \in \mathcal{K}} \left| \bar{\lambda}_k^T - \gamma^S \bar{\lambda}_k^S \right| \quad (10) \\ \leq \sum_{i \in T} \min_{j \in S} \left( \left\| \mathbf{g}_i^T - \mathbf{g}_j^S \right\|_2 + \frac{1}{2} \rho \sum_{k \in \mathcal{K}} \left| \hat{\lambda}_{i,k}^T - \hat{\lambda}_{j,k}^S \right| \right) \end{aligned}$$

Here,  $\bar{\mathbf{g}}^T = \frac{1}{|T|} \sum_{(x_i, y_i) \in T} \mathbf{g}_i^T$ <sup>4</sup> and  $\bar{\lambda}_k^T = \frac{1}{|T|} \sum_{(x_i, y_i) \in T} \hat{\lambda}_{i,k}^T$ . We aim at minimizing the upper bound from Eq (10), where we denote the upper bound as  $L(S)$ . Finally, our algorithm

will be implemented as follows:

$$\min_{S \subseteq T} L(S) \quad \text{s.t.} \quad |S| = m \quad (11)$$

Similar to Pooladzandi et al. (2022), we re-formulate Eq (11) into the formalized version of facility location algorithm. Let's suppose an auxiliary example  $e$ , and the minimization of  $L(S)$  is turned into the maximization of a facility location objective  $F(S)$  as follows:

$$\max_{S \subseteq T} F(S) = L(\{e\}) - L(S \cup \{e\}) \quad \text{s.t.} \quad |S| = m \quad (12)$$

Here,  $L(\{e\})$  is a constant, which is an upper bound of  $L(S)$ . The objective could also be derived as a submodular cover problem, whose objective is to minimize  $|S|$  with the constraints on  $F(S)$ . Finally, we call our method applied to the selection-based method as LCMat-S.<sup>5</sup>

### 3.4 LCMat for Condensation-based method

Different from selection-based methods, which need submodular optimization for a subset selection from  $T$ ; condensation-based methods directly optimize  $S$  by setting Eq (7) to  $\mathcal{L}(T, S; \theta)$ . Eventually, the implemented objective becomes  $\min_S \mathcal{L}(T, S; \theta)$ . Here,  $S$  is updated as  $S \leftarrow S - \gamma \nabla_S \mathcal{L}(T, S; \theta)$ . However, the direct optimization of Eq (7) still remains costly because of derivative computation over the Hessian terms, which are  $\hat{\lambda}_k^T$  and  $\hat{\lambda}_k^S$ . This section provides an efficient variation of Eq (7), which is adapted to the community of condensation-based methods.

According to Rame et al. (2022), the Fisher information  $\mathbb{F} = \sum_{i=1}^{|T|} \mathbb{E}_{\hat{y} \sim p_{\theta}(\hat{y}|x_i)} \left[ \nabla_{\theta} \log p_{\theta}(\hat{y}|x_i) \nabla_{\theta} \log p_{\theta}(\hat{y}|x_i)^{\top} \right]$  approximates the Hessian  $\mathbb{H}$  with probably bounded errors under mild assumptions (Kim et al., 2022b). As the Fisher information only requires the first derivative on  $\theta$ , the computation of Fisher information is more efficient than the computation of the Hessian matrix. The equation below is the empirical Fisher information  $\tilde{\mathbb{F}}$  of a certain dataset  $D$ :

$$\tilde{\mathbb{F}} = \frac{1}{|D|} \sum_{(x_i, y_i) \in D} \nabla_{\theta} \ell(x_i, y_i; \theta) \nabla_{\theta} \ell(x_i, y_i; \theta)^{\top} \quad (13)$$

$\tilde{\mathbb{F}}$  is equivalent<sup>6</sup> to the gradient covariance matrix  $\mathbf{C} = \frac{1}{n-1} \left( \mathbf{G}^{\top} \mathbf{G} - \frac{1}{n} (\mathbf{1}^{\top} \mathbf{G})^{\top} (\mathbf{1}^{\top} \mathbf{G}) \right)$  of size  $|\theta| \times |\theta|$  at any first-order stationary point (Rame et al., 2022), where  $\mathbf{G} = [\mathbf{g}_i]_{i=1}^{|D|}$ . As our objective (7) is constructed based on the Hessian diagonals, such as  $\hat{\mathbb{H}}_T$  and  $\hat{\mathbb{H}}_S$ ; we consider the gradient variance,  $\operatorname{Var}(\mathbf{G})$ , which is the diagonal components of  $\mathbf{C}$  as follows:

$$\operatorname{Var}(\mathbf{G}) = \frac{1}{|D| - 1} \sum_{i=1}^{|D|} (\mathbf{g}_i - \bar{\mathbf{g}})^2 \quad (14)$$

<sup>5</sup>The code is available at <https://github.com/SJShin-AI/LCMat>.

<sup>6</sup>We skip the index with  $D$  for the simplicity of  $\tilde{\mathbb{F}}$ ,  $\mathbf{C}$ , and  $\mathbf{G}$ .

<sup>4</sup> $\bar{\mathbf{g}} = \frac{1}{N} \sum_{i=1}^N \mathbf{g}_i$ .

Results from Rame et al. (2022) support that the similarity between Hessian diagonals and gradient variances is over 99.99%. Similar to Eq (9), we could specify  $\mathcal{K}$  to select the sub-dimensions of  $\text{Var}(\mathbf{G})$  to match. In practice, we match the whole dimensions of  $\text{Var}(\mathbf{G})$ , which shows the robustness over the implemented experiments. We provide the adapted application of LCMat to the dataset condensation as follows:

$$\begin{aligned} \min_S \mathbb{E}_{\theta^0} \left[ \sum_k \mathcal{D}(\bar{\mathbf{g}}_{\theta_k}^T, \bar{\mathbf{g}}_{\theta_k}^S) + \frac{1}{2} \rho |\text{Var}(\mathbf{G}_{\theta_k}^T) - \text{Var}(\mathbf{G}_{\theta_k}^S)| \right] \\ \text{s.t. } \theta_{t+1} = \theta_t - \eta \bar{\mathbf{g}}_{\theta_t}^T \text{ for } t = 0, \dots, k-1. \end{aligned} \quad (15)$$

We denote  $\theta_k$  under each term to represent the subject of the derivative. Our objective is composed of 1)  $\mathcal{D}(\bar{\mathbf{g}}_{\theta_k}^T, \bar{\mathbf{g}}_{\theta_k}^S)$ , which is averaged gradient matching between  $T$  and  $S$ ; and 2)  $|\text{Var}(\mathbf{G}_{\theta_k}^T) - \text{Var}(\mathbf{G}_{\theta_k}^S)|$ , which is gradient variance matching between  $T$  and  $S$ . Note that the averaged gradient matching is the objective of Zhao et al. (2020). We also differentiate the learning trajectory of  $\theta$  from  $S$  to  $T$  to satisfy the assumption on the model parameter in Section 3.5, which is utilized for the theoretical analysis of our method. We call our method applied to the condensation-based method as LCMat-C.

### 3.5 Theoretical Understanding of LCMat

This section analyzes the generalization bound of Eq (4), which is our primary objective. First, we define  $\hat{\Theta}$ , which is the application range of generalization bound as follows:

**Definition 3.3.**  $\hat{\Theta} = \{\theta : \mathcal{L}(T; \theta) \leq \mathcal{L}(\mathbb{D}; \theta) \text{ for } \theta \in \Theta\}$

In practice,  $\mathcal{L}(T; \theta)$  and  $\mathcal{L}(\mathbb{D}; \theta)$  are approximated by the training loss and test loss, respectively.  $\hat{\Theta}$  specifies  $\theta$  whose generalization gap is more than equal to zero, which is intuitive when we optimize  $\theta$  based on  $T$ . We first derive the generalization bound of  $\max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{abs}(T, S; \theta + \epsilon)$ , which is subpart of Eq (4), as follows:

**Theorem 3.4.** (Generalization Bound of  $\max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{abs}(T, S; \theta + \epsilon)$ ) For  $\theta \in \hat{\Theta}$ , with probability at least  $1 - \delta$  over the choice of the training set  $T$  with  $|T| = n$ , the following holds. (Proof in Appendix A.4)

$$\begin{aligned} \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} [\mathcal{L}_{abs}(\mathbb{D}, S; \theta + \epsilon)] \\ \leq \max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{abs}(T, S; \theta + \epsilon) + \sqrt{\frac{O(k + \log \frac{n}{\delta})}{n-1}} \end{aligned} \quad (16)$$

Please note that proof of Theorem 3.4 largely referred to the proof concept of SAM (Foret et al., 2020). Having said that, Theorem 3.4 states that  $\max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{abs}(T, S; \theta + \epsilon)$  can become the upper bound of  $\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} [\mathcal{L}(\mathbb{D}; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)]$ , which is the expected loss difference between  $\mathbb{D}$  and  $S$  over the  $\epsilon$ -perturbed space of the current parameter  $\theta$ .

From the theoretical view, Theorem 3.4 provides the generalization property of the loss difference between two arbitrary datasets. As an extension of Theorem 3.4, Corollary 1 directly investigates the generalization property of our main objective in Eq (4), which is the first term in R.H.S of Corollary 1, with an additional assumption,  $\mathcal{L}_{abs}(T, S; \theta) \leq \mathcal{L}_{abs}(\mathbb{D}, S; \theta)$ . The assumption is acceptable if the loss difference from  $\mathbb{D}$  is larger than  $T$ 's.

**Corollary 3.5.** (Generalization Bound of Eq (4)) If  $\mathcal{L}_{abs}(T, S; \theta) \leq \mathcal{L}_{abs}(\mathbb{D}, S; \theta)$  for  $\theta \in \hat{\Theta}$ , with probability at least  $1 - \delta$  over the choice of the training set  $T$  with  $|T| = n$ , the following holds: (Proof in Appendix A.5)

$$\begin{aligned} \left( \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} [\mathcal{L}_{abs}(\mathbb{D}, S; \theta + \epsilon)] - \mathcal{L}_{abs}(\mathbb{D}, S; \theta) \right) / \rho \\ \leq \left( \max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{abs}(T, S; \theta + \epsilon) - \mathcal{L}_{abs}(T, S; \theta) \right) / \rho \\ + \sqrt{\frac{O(k + \log \frac{n}{\delta})}{n-1}} \end{aligned} \quad (17)$$

According to Corollary 3.5, Eq (4) can be an upper bound of  $\left( \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} [\mathcal{L}_{abs}(\mathbb{D}, S; \theta + \epsilon)] - \mathcal{L}_{abs}(\mathbb{D}, S; \theta) \right) / \rho$ , which is the expected sharpness of loss differences between  $\mathbb{D}$  and  $S$  over the  $\epsilon$ -perturbed space of the parameter  $\theta$ . This implies that the minimization of Eq (4) would lead to the local curvature matching between  $S$  and  $\mathbb{D}$ , when  $\mathbb{D}$  is our target population distribution.

## 4 EXPERIMENTS

This section investigates the validity of our method, LCMat, through experiments on various datasets and tasks. First, we check the efficacy of LCMat through the application of LCMat on coreset selection and dataset condensation tasks. In addition, we investigate the performance of LCMat on a continual learning framework as a practical application.

### 4.1 Coreset Selection Evaluation

**Experiment Details** To investigate the efficacy of each selection-based algorithm, we follow the selection evaluation scenario of Guo et al. (2022), which is provided as follows. Each selection-based method learns  $S$  by utilizing the neural network,  $f_{\theta_T}$ , which is pre-trained on  $T$ . Next, we introduce another randomly initialized neural network  $f_{\theta_S}$ ; and we optimize  $\theta_S$  with  $S$ . Finally, we measure the test accuracy on  $f_{\theta_S}$  to evaluate the quality of  $S$ . During the selection, we assume that  $\theta$  is fixed without alternative optimization between  $S$  and  $\theta$ . It should be noted that our method could also be evaluated on the dynamic coreset selection scenario (Mirzasoileiman et al., 2020; Pooladzandi et al., 2022).

Table 1: Test accuracies of Coreset selection tasks on CIFAR-10 and CIFAR-100 with the deployment of ResNet-18 over 5 different random seeds. The best results and second-bests from each setting are shown in **bold** and underlines, respectively.

Fraction	CIFAR-10								CIFAR-100							
	0.1%	0.5%	1%	5%	10%	20%	30%	100%	0.5%	1%	5%	10%	20%	30%	100%	
Uniform	20.42±2.0	31.98±1.9	36.47±1.9	<u>64.21±2.1</u>	<b>77.45±1.0</b>	<u>87.36±0.4</u>	90.67±0.2		5.04±0.5	8.70±0.5	<u>25.37±0.3</u>	34.09±2.4	55.98±0.7	<u>64.59±0.1</u>		
C-Div	16.26±2.2	20.97±2.3	23.50±2.8	40.25±1.3	56.85±1.7	83.24±1.7	90.93±0.5		4.76±0.1	6.01±0.5	13.62±0.5	20.53±0.6	44.91±1.9	58.60±2.7		
Herdling	18.34±2.1	28.64±1.5	31.91±3.8	48.38±2.6	63.04±2.5	73.24±1.8	79.93±1.5		4.42±0.2	6.93±0.2	18.24±1.6	26.47±0.2	42.83±1.9	52.14±1.4		
k-Center	19.38±0.7	25.80±1.1	31.61±1.1	55.55±2.1	72.12±1.7	86.79±0.6	90.83±0.3		4.76±0.3	6.74±0.8	18.41±0.4	27.37±1.5	52.1±0.8	63.74±0.7		
L-Conf	13.67±2.0	18.05±1.4	20.31±1.8	36.14±2.2	58.43±3.0	82.64±1.2	<u>91.21±0.1</u>		2.65±0.1	4.38±0.1	11.31±0.4	17.63±2.1	41.29±1.1	58.86±1.0		
Entropy	15.29±1.1	17.50±2.0	22.42±2.0	37.92±2.4	57.45±3.6	81.72±2.2	91.06±0.7		2.51±0.4	3.82±0.3	11.32±0.5	16.94±0.9	41.88±1.3	57.45±2.0		
Margin	17.80±2.1	24.64±1.2	28.26±2.9	44.17±2.8	59.90±6.7	82.34±0.9	90.92±0.4		3.86±0.3	6.11±0.2	14.57±0.2	20.70±1.1	46.36±2.7	59.45±2.2		
Craig	18.80±2.4	27.40±1.9	29.76±2.0	39.75±3.7	51.73±4.6	74.09±0.9	87.25±0.8	95.48±0.1	6.38±0.4	<u>9.07±0.2</u>	15.93±0.4	20.32±0.6	32.23±0.2	47.09±1.4		
GradMatch	15.31±0.6	23.88±1.2	27.78±2.0	40.75±3.1	51.11±2.3	71.84±3.5	84.88±1.4		4.28±0.4	6.26±0.5	14.19±1.1	20.23±0.5	40.28±1.1	51.03±1.5		
GradMatch-Val	15.39±1.2	22.18±1.1	25.1±1.7	37.76±1.2	49.21±2.4	71.14±1.7	83.34±1.4		4.43±0.5	5.57±0.2	13.45±0.6	22.99±0.6	39.84±2.0	51.72±1.8		
Glistler	19.08±2.1	26.35±1.7	29.46±3.4	40.74±3.1	56.89±2.7	78.27±0.5	89.73±0.4		4.22±0.4	6.46±0.7	16.49±0.5	24.07±0.4	44.42±1.4	56.81±1.2		
Glistler-Val	17.53±1.2	23.97±0.8	28.64±1.7	39.74±1.1	52.98±2.1	77.54±2.3	87.46±1.1		4.54±0.2	5.5±0.5	14.78±1.1	25.72±1.0	43.22±1.0	55.98±1.2		
AdaCore	<u>22.54±0.9</u>	<u>32.02±1.1</u>	<u>39.09±1.0</u>	63.97±1.1	76.44±1.5	87.21±0.2	90.54±0.4		5.43±0.2	7.96±0.2	23.96±1.0	<u>35.26±1.8</u>	<u>56.54±0.6</u>	64.06±0.9		
LCMat-S	<b>23.87±1.1</b>	<b>33.17±0.6</b>	<b>39.54±0.7</b>	<b>64.72±1.3</b>	<u>77.41±2.0</u>	<b>88.12±0.2</b>	<b>91.32±0.2</b>		<b>7.65±0.8</b>	<b>11.82±0.8</b>	<b>27.3±1.2</b>	<b>36.66±1.0</b>	<b>56.66±0.6</b>	<b>64.81±0.9</b>		

**Baselines** We choose the baselines in the past works of selection-based methods. The selected baselines can be divided into two modelling categories. Baselines in the first category utilize the output from the forward-pass of the model, e.g. layer-wise feature vector, softmax output (Contextual Diversity (C-Div) (Agarwal et al., 2020), Herdling (Welling, 2009), k-CenterGreedy (k-Center) (Sener and Savarese, 2018), Least Confidence (L-Conf), Entropy, and Margin (Coleman et al., 2019)). Baselines in another category are a set of variants for gradient matching (Craig (Mirzasoleiman et al., 2020), GradMatch (Killamsetty et al., 2021a), Glistler (Killamsetty et al., 2021b) and AdaCore (Pooladzandi et al., 2022)). We also report results from a randomly chosen subset (Uniform). For all methods, We select  $S$  in a class-balanced manner. We provide the detailed implementation of each method and the corresponding wall-clock time in Appendix D.

**Implementation of LCMat-S and Gradient-based Methods** We compute the gradient and the Hessian matrix of the last layer of  $f_\theta$ , which is common practice in the theoretical analyses (Mirzasoleiman et al., 2020; Pooladzandi et al., 2022). For AdaCore (Pooladzandi et al., 2022) and our method, LCMat-S; we skip the training of  $w$ , which is learnable weights for the instances in subset  $S$  because it significantly decreases the test performances. We conjecture that the problem is caused by the over-fitting of  $w$ . We tune  $\rho$ , which is the only hyper-parameter of LCMat-S, from the value list of [0.01, 0.05, 0.1, 0.5]. We also implement the variants of GradMatch and Glistler, which we call as GradMatch-Val and Glistler-Val, by matching the gradient of  $T$  with the gradient over the validation dataset as specified in the original paper.

**Benchmark Evaluation Result** Table 1 reports the test accuracy of the ResNet-18 trained using  $S$  from each method. We evaluate  $S$  with different fractions in dataset reduction, which is the cardinality budget of  $S$  from  $T$ . Uniform, which is a random selection baseline, shows competitive

performances over other baselines. This shows the weak robustness of the existing selection methods. LCMat-S shows the improved or competitive performances over the implemented baselines by relieving the over-fitting issue of  $S$  to the provided  $\theta$ . Particularly, the gain from LCMat-S becomes significant when the tested dataset becomes difficult and the reduction rate becomes small, i.e. the dataset reduction to 0.5%, 1%, and 5% in CIFAR-100. In Appendix D.4, we report image samples selected by each method of all classes for CIFAR-10 dataset. LCMat-S selects a set of examples with diverse characteristics, e.g. the diverse shape of each object and different backgrounds without redundancy.

Table 2: Cross-architecture generalization performance (%) on CIFAR-100 with ResNet-18. **Bold** represents best result. Experiments are repeated over 3 times.

Fraction	Test Model	ResNet-18	VGG-16	Inception-v3	WRN-16-8
1%	Uniform	8.35±0.4	3.5±1.1	6.22±0.3	8.57±0.2
	Craig	9.65±0.3	2.53±0.6	6.07±0.5	10.37±0.2
	GradMatch	6.72±0.2	2.11±0.6	4.70±0.5	7.14±0.2
	Glistler	6.66±0.1	3.98±0.6	5.24±0.2	6.96±0.4
	AdaCore	7.85±0.1	2.53±0.6	5.88±0.3	8.61±0.1
	LCMat-S	<b>12.17±0.1</b>	<b>5.09±1.0</b>	<b>9.04±0.2</b>	<b>12.53±0.2</b>
5%	Uniform	25.85±0.0	18.22±0.8	21.00±0.4	30.13±0.5
	Craig	17.08±0.6	10.00±0.7	12.11±1.2	18.85±0.4
	GradMatch	15.63±0.0	12.59±0.2	13.43±0.2	19.16±0.7
	Glistler	17.01±0.3	13.82±0.7	14.14±0.3	20.53±0.7
	AdaCore	24.71±0.4	19.38±1.2	21.66±0.9	29.77±1.2
	LCMat-S	<b>27.29±0.7</b>	<b>20.42±1.2</b>	<b>24.87±0.7</b>	<b>33.20±0.7</b>

**Robustness on Cross-Architecture** From our scenario, the network structure of  $f_{\theta_S}$  could be different from  $f_{\theta_T}$ . We test the robustness of LCMat-S on the specific scenario, which we call as Cross-Architecture Generalization (Zhao et al., 2020). We utilize VGG-16 (Simonyan and Zisserman, 2014), Inception-v3 (Szegedy et al., 2016), and WRN-16-8 (Zagoruyko and Komodakis, 2016) as  $f_{\theta_S}$ . Table 2 reports the test accuracy of LCMat-S and other gradient-based methods. LCMat-S consistently shows better generalization performances than the implemented baselines. We conjecture



Table 3: Test accuracies of coreset selection task on VGG-16 network (first, second row) and TinyImageNet dataset (third row), respectively. We denote the best performance as **Bold**; and the second best performance as Underline, respectively.

Experiment	Frac	Uniform	k-Center	Craig	GradMatch	Gliester	AdaCore	LCMat-S
CIFAR-10 w/ VGG-16	0.5%	13.61±1.8	12.81±1.1	<b>15.83±1.9</b>	11.33±0.6	12.4±0.7	13.84±1.6	<u>15.37±0.0</u>
	1%	19.81±2.4	15.78±4.1	15.19±1.4	13.7±1.8	<u>22.84±3.1</u>	19.08±8.4	<b>25.41±6.4</b>
CIFAR-100 w/ VGG-16	0.5%	1.85±0.4	1.51±0.2	2.13±0.6	<b>2.41±0.8</b>	2.03±0.6	1.79±0.3	<u>2.34±0.2</u>
	1%	3.6±1.5	2.07±0.6	<u>4.73±1.0</u>	2.63±0.5	4.36±1.1	2.9±0.8	<b>5.91±0.3</b>
TinyImageNet w/ ResNet-18	0.5%	2.07±0.2	1.72±0.2	<u>2.99±0.2</u>	2.44±0.2	2.75±0.0	1.81±0.1	<b>3.18±0.4</b>
	1%	3.57±0.1	2.45±0.2	5.16±0.3	4.81±0.1	<u>5.20±0.3</u>	3.43±0.1	<b>5.43±0.4</b>

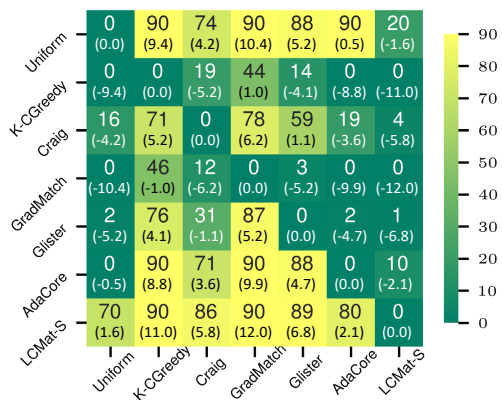


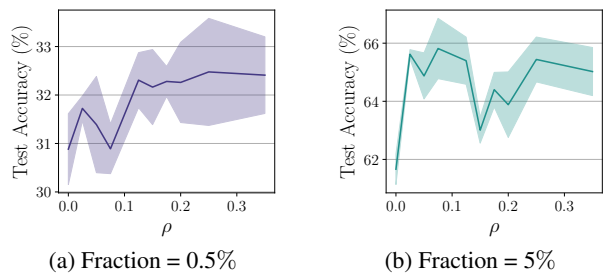
Figure 3: Heatmap plot which shows the number of times that each method beats the others from each case; and the averaged improvements over the other methods in parenthesis (%).

that the robustness over the different network architectures could be improved from our loss-curvature matching objective.

**Robustness on the pre-training of  $f_{\theta_T}$**  From our evaluation scenario,  $f_{\theta_T}$  could be pre-trained with different hyper-parameters for each experiment, where  $f_{\theta_T}$  significantly influences the selection of  $S$ . To test the robustness over the  $\theta$  pre-training, We conduct the coreset selection experiments over the differently pre-trained ResNet-18 with combinations of epochs [2,5,10,20,100]; weight decay [1e-4, 5e-4, 1e-3]; optimizers [SGD, Adam]; and 3 seeds, which result in 90 cases. Fig 3 shows the number of times that each method beats the others from each case; and the averaged improvements over the other methods in parenthesis (%). LCMat-S beats other baselines with large numbers.

**Additional Results** We demonstrate the efficacy of LCMat-S over the baselines from the experiments of 1) Selection with different network architecture (VGG-16); and 2) Selection on a different dataset (TinyImageNet). Table 3 shows that LCMat is consistently competitive over the selected baselines on the evaluated settings.

**Ablation Study** When we set  $\rho = 0$  in Eq (11), our method is reduced to the gradient matching with the facility location algorithm. To validate the efficacy of loss-curvature match-

Figure 4: The sensitivity analyses based on  $\rho$  for fraction = 0.5% and 5% in CIFAR-10.

ing over the gradient matching, we provide the ablation study of LCMat-S by conducting sensitivity analyses over  $\rho$ . Figure 4 shows that the test performances when  $\rho > 0$  are consistently higher than when  $\rho = 0$ , which shows the efficacy of loss-curvature matching over the gradient matching.

## 4.2 Dataset Condensation Evaluation

**Experiment Details** The condensation evaluation scenario is very similar to our selection scenario explained in Section 4.1. The only difference is the existence of alternative training between  $S$  and  $\theta_T$  during the condensation, which arises from the nature of condensation.

We condense  $S$  based on CIFAR-10 and CIFAR-100 with the utilization of ConvNet-3 as  $f_{\theta_T}$ . As specified in Eq (15), we optimize  $\theta_T$  from  $T$  but the current  $S$  during the alternative training of  $S$  and  $\theta_T$ , which is shown to be effective for the condensation (Kim et al., 2022a). It also fits with our parameter coverage on the Theorem 3.4. All methods utilize the Differential Siamese Augmentation (Zhao and Bilen, 2021a) and the additional augmentation strategy specified in Kim et al. (2022a) to further improve the performance. During the alternative update of  $S$  and  $\theta_T$ , we re-initialize  $\theta_T$  periodically as a common practice (Zhao et al., 2020; Kim et al., 2022a). All experiments in this section are repeated over 3 times.

**Baselines** To validate the efficacy of LCMat-C, we compare the test performances over the baselines with different objec-



Table 4: Condensation performances on CIFAR-10 and CIFAR-100 with ConvNet-3. **Bold** represents best result. † means reported results from the original papers.

Fraction	CIFAR-10		CIFAR-100	
	0.2%	1%	2%	10%
Random	37.13±0.3	56.67±0.5	20.60±0.3	40.90±0.0
KIP	47.30 <sup>†</sup> ±0.3	50.10 <sup>†</sup> ±0.2	13.40 <sup>†</sup> ±0.2	-
DM	54.47±0.5	65.23±0.2	33.99±0.2	43.35±0.2
DSA	54.90±0.3	61.90±0.4	33.75±0.1	38.71±0.3
LCMat-C	<b>56.83</b> ±0.2	<b>65.90</b> ±0.4	<b>36.47</b> ±0.0	<b>43.53</b> ±0.1
Full	89.77±0.2		65.13±0.5	

tives. Baselines include the gradient matching (DSA) (Zhao and Bilen, 2021a), feature output matching (DM) (Zhao and Bilen, 2021b), and kernel-based (KIP) methods (Nguyen et al., 2021).

**Implementation of LCMat-C** The gradient variance,  $\text{Var}(\mathbf{G}_{\theta_k}^T)$ , in Eq (15), requires the costly computation of per-sample gradients over  $\theta$ . We utilize BackPACK (Dangel et al., 2020), which provides the computation of per-sample gradients at almost no time overhead. Also, we compute the gradient variance term only for the last layer, which is an efficient practice to improve the test performance with low computational costs.

**Results** Table 4 shows that condensed  $S$  from LCMat-C consistently improves the test performances of all baselines over different fractions of CIFAR-10 and CIFAR-100. We especially observe significant improvements from the experiments on the low fraction budgets. We also test the robustness of LCMat-C on the cross-architecture scenario, which utilizes ResNet-10 (He et al., 2016) and DenseNet-121 (Huang et al., 2017) as testing backbones. Table 5 shows the consistent improvements of LCMat-C over baselines.

Table 5: Cross-architecture generalization performance (%) on CIFAR-10 with ConvNet-3. **Bold** means best.

Fraction	Test Model	ConvNet-3	ResNet-10	DenseNet-121
0.2%	Random	37.13±0.3	35.27±0.4	36.93±0.6
	DM	54.47±0.5	44.73±1.1	44.97±0.3
	DSA	54.90±0.3	46.03±0.3	45.63±1.8
	LCMat-C	<b>56.83</b> ±0.2	<b>48.00</b> ±1.5	<b>47.27</b> ±1.1
1%	Random	56.67±0.5	53.57±0.4	56.77±0.4
	DM	65.23±0.2	56.77±0.1	55.80±0.4
	DSA	61.90±0.4	57.97±0.2	55.00±0.8
	LCMat-C	<b>65.90</b> ±0.4	<b>60.93</b> ±0.4	<b>57.93</b> ±0.1
100%	Full	89.72±0.2	93.80±0.3	96.17±0.2

### 4.3 Application : Continual Learning with Memory Replay

Methods for memory-based continual learning store small representative instances; and these methods optimize its

classifier with the samples stored in the memory to alleviate the catastrophic forgetting of previously observed tasks (Chaudhry et al., 2019). As an application practice, we utilize  $S$  from each method as a memory exemplar for previously seen classes under the class incremental setting of Zhao et al. (2020); Zhao and Bilen (2021b). From the setting, CIFAR-100 is divided into 5 sets of sub-classes with a memory budget of 10 images per class, where each set of classes means a separate task stage. This setting purely trains a model based on the latest memory at each task stage. Figure 5 shows that the variants of LCMat, LCMat-S and LCMat-C, significantly improve the test performances under the defined setting, which represents the minimization of catastrophic forgetting.

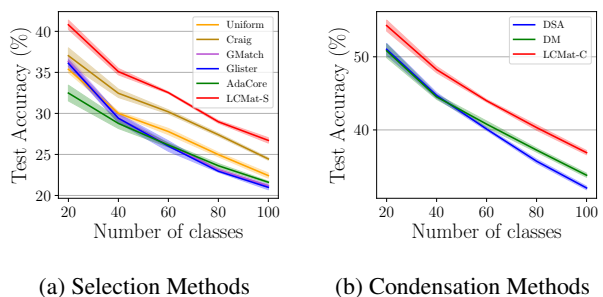


Figure 5: Test accuracy from the continual learning scenario with the selected or condensed data from CIFAR-100. We compare the methods from each category separately.

## 5 CONCLUSION

We propose a new objective for dataset reduction named Loss-Curvature Matching, or LCMat. LCMat identifies the worst loss-curvature gap between the original dataset and the reduced dataset around the local parameter region, which is closely related to the parameter-based generalization on dataset reduction procedure. From the adaptive application of LCMat, such as selection-based methods and condensation-based methods; LCMat consistently shows improved performances over baselines from both lines of research in dataset reduction. Especially, LCMat shows clear performance merits on the extreme reduction ratio, which is a specialized property for on-device learning where the memory capacity is limited.

### Acknowledgements

This research was supported by AI Technology Development for Commonsense Extraction, Reasoning, and Inference from Heterogeneous Data (IITP) funded by the Ministry of Science and ICT(2022-0-00077). Also, authors would like to acknowledge Dongjun Kim and Byeonghu Na for their invaluable discussions and supports.

## References

- Agarwal, S., Arora, H., Anand, S., and Arora, C. (2020). Contextual diversity for active learning. In *European Conference on Computer Vision*, pages 137–153. Springer.
- Borsos, Z., Mutny, M., and Krause, A. (2020). Coresets via bilevel optimization for continual learning and streaming. *Advances in Neural Information Processing Systems*, 33:14879–14890.
- Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P. K., Torr, P. H., and Ranzato, M. (2019). On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*.
- Coleman, C., Yeh, C., Mussmann, S., Mirzasoleiman, B., Bailis, P., Liang, P., Leskovec, J., and Zaharia, M. (2019). Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations*.
- Dangel, F., Kunstner, F., and Hennig, P. (2020). BackPACK: Packing more into backprop. In *International Conference on Learning Representations*.
- Dong, T., Zhao, B., and Lyu, L. (2022). Privacy for free: How does dataset condensation help privacy? *arXiv preprint arXiv:2206.00240*.
- Ducoffe, M. and Precioso, F. (2018). Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*.
- Dziugaite, G. K. and Roy, D. M. (2017). Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*.
- Elenberg, E. R., Khanna, R., Dimakis, A. G., and Negahban, S. (2018a). Restricted strong convexity implies weak submodularity. *The Annals of Statistics*, 46(6B):3539–3568.
- Elenberg, E. R., Khanna, R., Dimakis, A. G., and Negahban, S. (2018b). Restricted strong convexity implies weak submodularity. *The Annals of Statistics*, 46(6B):3539–3568.
- Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. (2020). Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*.
- Guo, C., Zhao, B., and 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*, pages 181–195. Springer.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- He, Z., Rakin, A. S., and Fan, D. (2019). Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 588–597.
- Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.
- Izmailov, P., Podoprikin, D., Garipov, T., Vetrov, D. P., and Wilson, A. G. (2018). Averaging weights leads to wider optima and better generalization. In Globerson, A. and Silva, R., editors, *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 876–885. AUAI Press.
- Killamsetty, K., Durga, S., Ramakrishnan, G., De, A., and Iyer, R. (2021a). Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pages 5464–5474. PMLR.
- Killamsetty, K., Sivasubramanian, D., Ramakrishnan, G., and Iyer, R. (2021b). Glister: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8110–8118.
- Kim, J.-H., Kim, J., Oh, S. J., Yun, S., Song, H., Jeong, J., Ha, J.-W., and Song, H. O. (2022a). Dataset condensation via efficient synthetic-data parameterization. In *International Conference on Machine Learning (ICML)*.
- Kim, M., Li, D., Hu, S. X., and Hospedales, T. (2022b). Fisher sam: Information geometry and sharpness aware minimisation. In *International Conference on Machine Learning*, pages 11148–11161. PMLR.
- Langford, J. and Caruana, R. (2001). (not) bounding the true error. In Dietterich, T., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press.
- Laurent, B. and Massart, P. (2000). Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338.
- Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018). Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31.
- Lin, H., Bilmes, J., and Xie, S. (2009). Graph-based submodular selection for extractive summarization. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 381–386. IEEE.
- Lin, H. and Bilmes, J. A. (2012). Learning mixtures of submodular shells with application to document summarization. *arXiv preprint arXiv:1210.4871*.

- Lopez-Paz, D. and Ranzato, M. (2017). Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30.
- Margatina, K., Vernikos, G., Barrault, L., and Aletras, N. (2021). Active learning by acquiring contrastive examples. In *EMNLP (1)*.
- McAllester, D. A. (1999). Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 164–170.
- Mirzasoleiman, B., Bilmes, J., and Leskovec, J. (2020). Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR.
- Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294.
- Nguyen, T., Chen, Z., and Lee, J. (2021). Dataset meta-learning from kernel ridge-regression. In *International Conference on Learning Representations*.
- Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., So, D., Texier, M., and Dean, J. (2021). Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*.
- Paul, M., Ganguli, S., and Dziugaite, G. K. (2021). Deep learning on a data diet: Finding important examples early in training. *Advances in Neural Information Processing Systems*, 34:20596–20607.
- Pooladzandi, O., Davini, D., and Mirzasoleiman, B. (2022). Adaptive second order coresets for data-efficient machine learning. In *International Conference on Machine Learning*, pages 17848–17869. PMLR.
- Rame, A., Dancette, C., and Cord, M. (2022). Fishr: Invariant gradient variances for out-of-distribution generalization. In *International Conference on Machine Learning*, pages 18347–18377. PMLR.
- Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. (2019). Distributionally robust neural networks. In *International Conference on Learning Representations*.
- Sener, O. and Savarese, S. (2018). Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Sun, X., Zhang, Z., Ren, X., Luo, R., and Li, L. (2021). Exploring the vulnerability of deep neural networks: A study of parameter corruption. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11648–11656.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.
- Toneva, M., Sordoni, A., des Combes, R. T., Trischler, A., Bengio, Y., and Gordon, G. J. (2018). An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*.
- Wang, K., Zhao, B., Peng, X., Zhu, Z., Yang, S., Wang, S., Huang, G., Bilen, H., Wang, X., and You, Y. (2022). Cafe: Learning to condense dataset by aligning features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12196–12205.
- Wang, T., Zhu, J.-Y., Torralba, A., and Efros, A. A. (2018). Dataset distillation. *arXiv preprint arXiv:1811.10959*.
- Welling, M. (2009). Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1121–1128.
- Wolf, G. W. (2011). Facility location: concepts, models, algorithms and case studies. series: Contributions to management science: edited by zanjirani farahani, reza and hekmatfar, masoud, heidelberg, germany, physica-verlag, 2009, 549 pp., € 171.15, 219.00, £144.00, isbn978-3-7908-2150-5(*hardprint*), 978-3-7908-2151-2(*electronic*).
- Wolsey, L. A. (1982). An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393.
- Wu, D., Xia, S.-T., and Wang, Y. (2020). Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969.
- Yao, Z., Gholami, A., Shen, S., Mustafa, M., Keutzer, K., and Mahoney, M. (2021). Adahessian: An adaptive second order optimizer for machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10665–10673.
- Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146*.
- Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. (2019). Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR.
- Zhao, B. and Bilen, H. (2021a). Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR.
- Zhao, B. and Bilen, H. (2021b). Dataset condensation with distribution matching. *CoRR*, abs/2110.04181.

Zhao, B., Mopuri, K. R., and Bilen, H. (2020). Dataset condensation with gradient matching. In *International Conference on Learning Representations*.

## A Proofs

### A.1 Proof of Proposition 1

**Proposition A.1.** When  $\mathbb{H}_D = \nabla_{\theta}^2 \mathcal{L}(D; \theta)$  is a Hessian matrix of  $\mathcal{L}(D; \theta)$ , let  $\mathbb{H}_{T,S} = \mathbb{H}_T - \mathbb{H}_S = \nabla_{\theta}^2 \mathcal{L}(T; \theta) - \nabla_{\theta}^2 \mathcal{L}(S; \theta)$  and  $\lambda_1^{T,S}$  be the maximum eigenvalue of the matrix  $\mathbb{H}_{T,S}$ , then we have:

$$\max_{\|\epsilon\|_2 \leq \rho} \frac{\mathcal{L}_{abs}(T, S; \theta + \epsilon) - \mathcal{L}_{abs}(T, S; \theta)}{\rho} \leq \underbrace{\left\| \nabla_{\theta} \mathcal{L}(T; \theta) - \nabla_{\theta} \mathcal{L}(S; \theta) \right\|_2}_{\text{Gradient Matching via } L_2\text{-norm.}} + \underbrace{\frac{1}{2} \rho \lambda_1^{T,S}}_{\text{Max eigen}} + \max_{\|v\|_2 \leq 1} O(\rho^2 v^3)$$

*Proof.* By leveraging the Taylor-series with finite-order approximation, we can derive  $\max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{abs}(T, S; \theta + \epsilon)$ , which is an abbreviated term introduced in Section 3 of main paper, as follows:

$$\begin{aligned} \max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{abs}(T, S; \theta + \epsilon) &= \max_{\|\epsilon\|_2 \leq \rho} \left| \mathcal{L}(T; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon) \right| \\ &= \max_{\|\epsilon\|_2 \leq \rho} \left| \mathcal{L}(T; \theta) + \epsilon^{\top} \nabla_{\theta} \mathcal{L}(T; \theta) + \frac{1}{2} \epsilon^{\top} \nabla_{\theta}^2 \mathcal{L}(T; \theta) \epsilon - \mathcal{L}(S; \theta) - \epsilon^{\top} \nabla_{\theta} \mathcal{L}(S; \theta) - \frac{1}{2} \epsilon^{\top} \nabla_{\theta}^2 \mathcal{L}(S; \theta) \epsilon + O(\epsilon^3) \right| \end{aligned} \quad (18)$$

$$\begin{aligned} &\leq |\mathcal{L}(T; \theta) - \mathcal{L}(S; \theta)| + \max_{\|\epsilon\|_2 \leq \rho} |\epsilon^{\top} (\nabla_{\theta} \mathcal{L}(T; \theta) - \nabla_{\theta} \mathcal{L}(S; \theta))| + \max_{\|\epsilon\|_2 \leq \rho} \left| \frac{1}{2} \epsilon^{\top} \nabla_{\theta}^2 \mathcal{L}(T; \theta) \epsilon - \frac{1}{2} \epsilon^{\top} \nabla_{\theta}^2 \mathcal{L}(S; \theta) \epsilon \right| \\ &+ \max_{\|v\|_2 \leq \rho} O(v^3) \\ &= |\mathcal{L}(T; \theta) - \mathcal{L}(S; \theta)| + \rho \|\nabla_{\theta} \mathcal{L}(T; \theta) - \nabla_{\theta} \mathcal{L}(S; \theta)\|^2 + \max_{\|\epsilon\|_2 \leq \rho} \left| \frac{1}{2} \epsilon^{\top} \nabla_{\theta}^2 \mathcal{L}(T; \theta) \epsilon - \frac{1}{2} \epsilon^{\top} \nabla_{\theta}^2 \mathcal{L}(S; \theta) \epsilon \right| + \max_{\|v\|_2 \leq 1} O(\rho^3 v^3) \end{aligned} \quad (19)$$

From here, we denote the difference of hessian,  $\mathbb{H}_{T,S} = \nabla_{\theta}^2 \mathcal{L}(T; \theta) - \nabla_{\theta}^2 \mathcal{L}(S; \theta)$ , and we derive  $\mathbb{H}_{T,S}$  as follows:

$$\begin{aligned} \max_{\|\epsilon\|_2 \leq \rho} \left| \frac{1}{2} \epsilon^{\top} \nabla_{\theta}^2 \mathcal{L}(T; \theta) \epsilon - \frac{1}{2} \epsilon^{\top} \nabla_{\theta}^2 \mathcal{L}(S; \theta) \epsilon \right| &= \max_{\|\epsilon\|_2 \leq \rho} \left| \frac{1}{2} \epsilon^{\top} \mathbb{H}_{T,S} \epsilon \right| = \max_{\|\epsilon\|_2 \leq \rho} \frac{1}{2} \|\epsilon\| \|\mathbb{H}_{T,S} \epsilon\| \\ &= \frac{1}{2} \rho^2 \max_{\|v\|_2 \leq 1} \|\mathbb{H}_{T,S} v\| = \frac{1}{2} \rho^2 \lambda_1^{T,S} \end{aligned} \quad (21)$$

Here,  $\lambda_i^{T,S}$  is maximum eigenvalue of the matrix  $\mathbb{H}_{T,S}$ . By replacing  $\lambda_i^{T,S}$  into the Eq (20),  $\max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{abs}(T, S; \theta + \epsilon)$  is derived as follows:

$$\begin{aligned} \max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{abs}(T, S; \theta + \epsilon) &= \max_{\|\epsilon\|_2 \leq \rho} \left| \mathcal{L}(T; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon) \right| \\ &\leq \mathcal{L}_{abs}(T, S; \theta) + \rho \|\nabla_{\theta} \mathcal{L}(T; \theta) - \nabla_{\theta} \mathcal{L}(S; \theta)\|^2 + \frac{1}{2} \rho^2 \lambda_1^{T,S} + \max_{\|v\|_2 \leq 1} O(\rho^3 v^3) \end{aligned} \quad (22)$$

After moving  $\mathcal{L}_{abs}(T, S; \theta)$  to L.H.S, dividing both sides by  $\rho > 0$  finishes the proof as follows:

$$\frac{\max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{abs}(T, S; \theta + \epsilon) - \mathcal{L}_{abs}(T, S; \theta)}{\rho} \leq \|\nabla_{\theta} \mathcal{L}(T; \theta) - \nabla_{\theta} \mathcal{L}(S; \theta)\|^2 + \frac{1}{2} \rho \lambda_1^{T,S} + \max_{\|v\|_2 \leq 1} O(\rho^2 v^3) \quad (23)$$

### A.2 Proof of Eq (7)

From the replacement of  $\mathbb{H}_T$  and  $\mathbb{H}_S$ ; into  $\hat{\mathbb{H}}_T$  and  $\hat{\mathbb{H}}_S$ , which are diagonal version of Hessian matrices of  $\mathbb{H}_T$  and  $\mathbb{H}_S$ , respectively, we could further simplify the derivation in Eq (21) of the supplementary material into following equations:

$$\begin{aligned} \max_{\|\epsilon\|_2 \leq \rho} \left| \frac{1}{2} \epsilon^{\top} \nabla_{\theta}^2 \mathcal{L}(T; \theta) \epsilon - \frac{1}{2} \epsilon^{\top} \nabla_{\theta}^2 \mathcal{L}(S; \theta) \epsilon \right| \\ = \frac{1}{2} \max_{\|\epsilon\|_2 \leq \rho} \left| \epsilon^{\top} (\hat{\mathbb{H}}_T - \hat{\mathbb{H}}_S) \epsilon \right| = \frac{1}{2} \rho \max_{\|\epsilon\|_2 \leq \rho} \left| \sum_k \epsilon_k (\hat{\lambda}_k^T - \hat{\lambda}_k^S) \right| = \frac{1}{2} \rho^2 \max_k \left| \hat{\lambda}_k^T - \hat{\lambda}_k^S \right| \end{aligned}$$

Here,  $\hat{\lambda}_k^T$  and  $\hat{\lambda}_k^S$  are eigenvalues of  $\hat{\mathbb{H}}_T$  and  $\hat{\mathbb{H}}_S$  on  $k$ -th parameter dimension for  $\theta$ .

### A.3 Proof of Eq (10)

In this section, we prove that Eq (10) with per-sample weight  $\gamma_j$  could be changed as follows:

$$\min_{S \subseteq T} \left\| \sum_{i \in T} \hat{\lambda}_{i, \mathcal{K}}^T - \sum_{j \in S} \gamma_j \hat{\lambda}_{j, \mathcal{K}}^S \right\| + \left\| \sum_{i \in T} \mathbf{g}_i^T - \sum_{j \in S} \gamma_j \mathbf{g}_j^S \right\| \leq \sum_{i \in T} \min_{j \in S} \left\| \mathbf{g}_i^T - \mathbf{g}_j^S \right\| + \left\| \hat{\lambda}_{i, \mathcal{K}}^T - \hat{\lambda}_{j, \mathcal{K}}^S \right\| \quad (24)$$

To derive the Eq (24), we first re-phrase the upper-bound derivation of Craig (Mirzasoleiman et al., 2020) with the notation based on our paper. Mirzasoleiman et al. (2020) showed that the norm-based error between sum of whole elements in  $T$  and a weighted sum of a subset of elements is upper-bounded by facility location objective. For the complete proof of Eq (10) in main paper, we also follow the proof of Mirzasoleiman et al. (2020) here. We assume that there is a mapping function  $\zeta_\theta(i) : T \rightarrow S$  which assigns every data point  $i \in T$  to one of the elements  $j \in S$ , i.e.  $\zeta_\theta(i) = j \in S$ . Corresponding set  $C_j = \{i \in [n] | \zeta(i) = j\} \subseteq T$  is defined as a set of data points that are assigned to  $j \in S$ , and  $\gamma_j = |C_j|$  be the number of samples assigned to  $j$ . From this derivation, we can write as follows:

$$\sum_{i \in T} \mathbf{g}_i^T = \sum_{i \in T} \left( \mathbf{g}_i^T - \mathbf{g}_{\zeta_\theta(i)}^T + \mathbf{g}_{\zeta_\theta(i)}^T \right) = \sum_{i \in T} \left( \mathbf{g}_i^T - \mathbf{g}_{\zeta_\theta(i)}^T \right) + \sum_{j \in S} \gamma_j \mathbf{g}_j^S \quad (25)$$

From above equation, subtracting  $\sum_{j \in S} \gamma_j \mathbf{g}_j^S$  and taking norm with triangle inequality, we get the upper bound as follows:

$$\left\| \sum_{i \in T} \mathbf{g}_i^T - \sum_{j \in S} \gamma_j \mathbf{g}_j^S \right\| \leq \sum_{i \in T} \left\| \mathbf{g}_i^T - \mathbf{g}_{\zeta_\theta(i)}^T \right\| \quad (26)$$

To construct the upper bound based on  $\hat{\lambda}_{i, k}^T$  and  $\hat{\lambda}_{j, k}^S$ , we first denote a vector  $\hat{\lambda}_{i, \mathcal{K}}^T = (\hat{\lambda}_{i, a}^T, \hat{\lambda}_{i, b}^T, \hat{\lambda}_{i, c}^T, \dots)$ , where  $a, b, c, \dots$  are naively introduced indices for a specific index  $k \in \mathcal{K}$ . Then for  $i, j \in T$ , following holds by definition.

$$\sum_{k \in \mathcal{K}} \left| \hat{\lambda}_{i, k}^T - \hat{\lambda}_{j, k}^S \right| = \left\| \hat{\lambda}_{i, \mathcal{K}}^T - \hat{\lambda}_{j, \mathcal{K}}^S \right\|_1 \quad (27)$$

As Eq (26) holds for any bounded vector, we could also extend the result of Eq (26) as follows:

$$\left\| \sum_{i \in T} \hat{\lambda}_{i, \mathcal{K}}^T - \sum_{j \in S} \gamma_j \hat{\lambda}_{j, \mathcal{K}}^S \right\| \leq \sum_{i \in T} \left\| \hat{\lambda}_{i, \mathcal{K}}^T - \hat{\lambda}_{\zeta_\theta(i), \mathcal{K}}^S \right\| \quad (28)$$

Integrating Eq (26) and (28),

$$\left\| \sum_{i \in T} \hat{\lambda}_{i, \mathcal{K}}^T - \sum_{j \in S} \gamma_j \hat{\lambda}_{j, \mathcal{K}}^S \right\| + \left\| \sum_{i \in T} \mathbf{g}_i^T - \sum_{j \in S} \gamma_j \mathbf{g}_j^S \right\| \leq \sum_{i \in T} \left\| \mathbf{g}_i^T - \mathbf{g}_{\zeta_\theta(i)}^T \right\| + \left\| \hat{\lambda}_{i, \mathcal{K}}^T - \hat{\lambda}_{\zeta_\theta(i), \mathcal{K}}^S \right\| \quad (29)$$

Then, we set the mapping function  $\zeta_\theta(i) = \operatorname{argmin}_{j \in S} \left\| \mathbf{g}_i^T - \mathbf{g}_j^S \right\| + \left\| \hat{\lambda}_{i, \mathcal{K}}^T - \hat{\lambda}_{j, \mathcal{K}}^S \right\|$ . Hence,

$$\min_{S \subseteq V} \left\| \sum_{i \in T} \hat{\lambda}_{i, \mathcal{K}}^T - \sum_{j \in S} \gamma_j \hat{\lambda}_{j, \mathcal{K}}^S \right\| + \left\| \sum_{i \in T} \mathbf{g}_i^T - \sum_{j \in S} \gamma_j \mathbf{g}_j^S \right\| \leq \sum_{i \in T} \min_{j \in S} \left\| \mathbf{g}_i^T - \mathbf{g}_j^S \right\| + \left\| \hat{\lambda}_{i, \mathcal{K}}^T - \hat{\lambda}_{j, \mathcal{K}}^S \right\| \quad (30)$$

Based on the upper bound, we provide the pseudo-code, which is the greedy algorithm of LCMat-S, in Algorithm 1. The provided pseudo-code is motivated from Pooladzandi et al. (2022). The notations are all defined from the main paper. As mentioned in the main paper, we report the performance of LCMat-S without the application of weighting procedure because the performance with weighting shows degraded performance than the performance without weights. It should be noted that the greedy algorithm only with the incremental selection procedure provides us a logarithmic approximation (Nemhauser et al., 1978; Wolsey, 1982).



---

**Algorithm 1** The Greedy Algorithm of LCMat-S
 

---

**Ensure:** Subset  $S \subseteq V$  with corresponding per-element stepsizes  $\{\gamma\}_{j \in S}$ .

```

1: procedure LCMAT-S
2:    $S_0 \leftarrow \emptyset, i = 0$ 
3:   while  $F(S) < C_1 - \epsilon$  do ▷ Selection Procedure of LCMat-S
4:      $j \in \arg \max_{e \in V \setminus S_{i-1}} F(e|S_{i-1})$ 
5:      $S_i = S_{i-1} \cup \{j\}$ 
6:      $i = i + 1$ 
7:   end while
8:   for  $j = 1$  to  $|S|$  do ▷ Weighting Procedure of LCMat-S (Optional)
9:      $\gamma_j = \sum_{i \in V} \mathbb{I} \left[ j = \arg \min_{j \in S} \left\| \mathbf{g}_i^T - \mathbf{g}_j^S \right\| + \left\| \hat{\lambda}_{i, \mathcal{K}}^T - \hat{\lambda}_{j, \mathcal{K}}^S \right\| \right]$ 
10:  end for
11: end procedure
    
```

---

#### A.4 Proof of Theorem 3.4

First, we define a set of  $\theta, \hat{\Theta}$ , which is the application range of generalization bound as follows:

**Definition A.2.**  $\hat{\Theta} = \{\theta : \mathcal{L}(T; \theta) \leq \mathcal{L}(\mathbb{D}; \theta) \text{ for } \theta \in \Theta\}$

As noted in the main paper,  $\mathcal{L}(T; \theta)$  and  $\mathcal{L}(\mathbb{D}; \theta)$  are approximated by the training loss and test loss from the experimental practices, respectively.  $\hat{\Theta}$  specifies  $\theta$  whose generalization gap is more than equal to zero, which is intuitive when we optimize  $\theta$  based on  $T$  based on the valid setting.

**Theorem A.3.** (Generalization Bound of  $\max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{abs}(T, S; \theta + \epsilon)$ ) For  $\theta \in \hat{\Theta}$ , with probability at least  $1 - \delta$  over the choice of the training set  $T$  with  $|T| = n$ , the following holds.

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} \left[ \left| \mathcal{L}(\mathbb{D}; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon) \right| \right] \leq \max_{\|\epsilon\|_2 \leq \rho} |\mathcal{L}(T; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)| + \sqrt{\frac{O(k + \log \frac{n}{\delta})}{n-1}}$$

*Proof.* We start the proof by utilizing the triangle inequality when each metric is provided by absolute difference as follows:

$$|x - z| \leq |x - y| + |y - z| \text{ for all } x, y, z \quad (31)$$

From the triangle inequality, we can derive the inequality between the losses from the different population as follows:

$$|\mathcal{L}(\mathbb{D}; \theta) - \mathcal{L}(S; \theta)| \leq |\mathcal{L}(\mathbb{D}; \theta) - \mathcal{L}(T; \theta)| + |\mathcal{L}(T; \theta) - \mathcal{L}(S; \theta)| \text{ for all } \theta \in \Theta \quad (32)$$

It can also be extended into the following inequality, which is inequality between the expected loss on the  $\epsilon$ -perturbed region of  $\theta$ :

$$\begin{aligned} \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} \left[ |\mathcal{L}(\mathbb{D}; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)| \right] &\leq \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} \left[ |\mathcal{L}(\mathbb{D}; \theta + \epsilon) - \mathcal{L}(T; \theta + \epsilon)| \right] \\ &+ \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} \left[ |\mathcal{L}(T; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)| \right] \text{ for all } \theta \in \Theta \end{aligned} \quad (33)$$

Here, we refer the PAC-Bayes theorem McAllester (1999) to derive the bound between them. It should be noted that the provided proof referred the proof concept of SAM Foret et al. (2020) and fisher-SAM (Kim et al., 2022b). The PAC-Bayes generalization bound of McAllester (1999); Dziugaite and Roy (2017) provides that, for any prior distribution  $P(\theta)$  with probability at least  $1 - \delta$  over the choice of the training set  $T$  with  $|T| = n$ , it holds that

$$\mathbb{E}_{Q(\theta)} [\mathcal{L}(\mathbb{D}; \theta)] \leq \mathbb{E}_{Q(\theta)} [\mathcal{L}(T; \theta)] + \sqrt{\frac{\text{KL}(Q(\theta) \| P(\theta)) + \log \frac{n}{\delta}}{2(n-1)}} \quad (34)$$

Posterior distribution,  $Q(\theta)$ , is assumed to be dependent on the training dataset  $T$  and synthetic data variable  $S$ , which are both accessible during the training procedure of  $\theta$ . Let  $k$  be the dimensionality of the model parameter  $\theta$ . Following

Kim et al. (2022b), if we assume that  $Q(\theta) = \mathcal{N}(\mu_Q, \sigma_Q^2 I)$  and  $P(\theta) = \mathcal{N}(\mu_P, \sigma_P^2 I)$ , the KL divergence can be written as follows:

$$\text{KL}(Q||P) = \frac{1}{2} \left[ \frac{k\sigma_Q^2 + \|\mu_Q - \mu_P\|_2^2}{\sigma_P^2} - k + k \log\left(\frac{\sigma_P^2}{\sigma_Q^2}\right) \right] \quad (35)$$

It should be noted that the prior distribution  $P(\theta)$  do not have access into the training dataset  $T$ , which makes it hard to adapt the  $P(\theta)$  to minimize KL-divergence with the corresponding posterior  $Q(\theta)$ . It inspires the utilization of the covering approach from Foret et al. (2020); Langford and Caruana (2001), which introduces a pre-defined set of parameter distributions with the constraint that each prior distribution holds the PAC-Bayes bound. Afterwards, we can select the one from the set which has minimal KL-divergence in junction with the posterior  $Q(\theta)$ .

From a pre-defined set of prior distributions  $\{P_j(\theta)\}_{j=1}^J$  where  $P_j(\theta) = \mathcal{N}(\bar{\theta}_j, \bar{\sigma}_j)$  and posterior distribution  $Q(\theta)$ , we set  $\mu_Q = \theta$ ,  $\sigma_Q = \rho$ , and  $\bar{\theta}_j = 0$ . Here, the point is how to set  $\bar{\sigma}_j$ . Motivated from Langford and Caruana (2001), we introduce  $\{c \exp((1-j)/k) | j \in \mathbb{N}\}$ , which is a set of pre-defined parameter values for  $\sigma_P^2$ . For the detailed analyses about the inclusion of  $c$ , see Langford and Caruana (2001) for the detailed explanation of the technique. From this setting, PAC-Bayes bound holds with probability  $1 - \delta_j$  when  $\delta_j = \frac{6\delta}{\pi^2 j^2}$ , which is generalized by the union bound theorem that all bounds hold simultaneously with probability at least  $1 - \sum_{j=1}^{\infty} \frac{6\delta}{\pi^2 j^2} = 1 - \delta$ .

With the specified  $Q(\theta)$  and  $P(\theta)$ , we have:

$$k\sigma_Q^2 + \|\mu_Q - \mu_P\|_2^2 = k\rho^2 + \|\theta\|_2^2 \quad (36)$$

With the replacement of  $Q(\theta)$  and  $P(\theta)$ , we rephrase Eq (34) as follows:

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} [\mathcal{L}_{\mathbb{D}}(\theta + \epsilon)] \leq \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} [\mathcal{L}_T(\theta + \epsilon)] + \sqrt{\frac{\frac{1}{4}k \left( \frac{\rho^2 + \|\theta\|_2^2/k}{\sigma_P^2} - 1 + \log \frac{\sigma_P^2}{\rho^2} \right) + \log \frac{n}{\delta}}{n-1}} \quad (37)$$

Here, we first restrict the value range of  $\sigma_P^2$  to further derive the bound of KL divergence. Afterwards, we provide that the specified value range is strictly feasible with some  $j \in \mathbb{N}$ . Having said that, we provide the range of  $\sigma_P^2$  as follows:

$$\rho^2 + \|\theta\|_2^2/k \leq \sigma_P^2 \leq \exp(1/k)(\rho^2 + \|\theta\|_2^2/k) \quad (38)$$

From the specified region of  $\sigma_P$ , KL divergence is bounded as follows:

$$\text{KL}(Q(\theta)||P(\theta)) = \frac{k}{2} \left( \frac{\rho^2 + \|\theta\|_2^2/k}{\sigma_P^2} - 1 + \log \frac{\sigma_P^2}{\rho^2} \right) \quad (39)$$

$$\leq \frac{k}{2} \left( \frac{\rho^2 + \|\theta\|_2^2/k}{\rho^2 + \|\theta\|_2^2/k} - 1 + \log \left( \frac{\exp(1/k)(\rho^2 + \|\theta\|_2^2/k)}{\rho^2} \right) \right) \quad (40)$$

$$= \frac{k}{2} \left( \frac{1}{k} + \log \left( 1 + \frac{\|\theta\|_2^2}{k\rho^2} \right) \right) \quad (41)$$

It should be noted that above bound holds only for specific  $j \in \mathbb{N}$ , which is not specified yet. Utilizing the provided bound of KL-divergence for specific  $j$  and subtracting the first term in R.H.S of Eq (37), it is further derived as follows:

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} [\mathcal{L}_{\mathbb{D}}(\theta + \epsilon) - \mathcal{L}_T(\theta + \epsilon)] \leq \sqrt{\frac{\frac{1}{4}k \log \left( 1 + \frac{\|\theta\|_2^2}{k\rho^2} \right) + \frac{1}{4} + \log \frac{n}{\delta_j}}{n-1}} \quad (42)$$

From the above bound, the value range of  $\|\theta\|_2^2$  divided into a range in which the bound holds trivially and a range in which it does not. The right hand side of (42) is lower-bounded by  $\sqrt{\frac{k}{4n} \log(1 + \|\theta\|_2^2/\rho^2)}$ , which is greater than 1 when  $\|\theta\|_2^2 \geq \rho^2(\exp(4n/k) - 1)$ . It guarantees that the right hand side of (42) is greater than 1, which results in the trivial proof of inequality. Having said that, we focus on the case when  $\|\theta\|_2^2 \leq \rho^2(\exp(4n/k) - 1)$ .

When  $\|\theta\|_2^2 \leq \rho^2(\exp(4n/k) - 1)$ , we have:

$$\rho^2 + \|\theta\|_2^2/k \leq \rho^2(1 + \exp(4n/k)) \quad (43)$$

By considering the bound where  $j = \lfloor 1 - k \log((\rho^2 + \|\theta\|_2^2/k)/c) \rfloor \in \mathcal{N}$  and setting  $c = \rho^2(1 + \exp(4n/k))$ , we can derive the feasible bound of  $\sigma_P^2$  as follows:

$$\rho^2 + \|\theta\|_2^2/k \leq \sigma_P^2 \leq \exp(1/k)(\rho^2 + \|\theta\|_2^2/k) \quad (44)$$

It is exactly same with the provided range of  $\sigma_P^2$  in Eq (38). The bound which corresponds to  $j$  holds with probability  $1 - \delta_j$  for  $\delta_j = \frac{6\delta}{\pi^2 j^2}$ . By leveraging it, we transform the log term  $\log \frac{n}{\delta_j}$  as follows:

$$\log \frac{n}{\delta_j} = \log \frac{n}{\delta} + \log \frac{\pi^2 j^2}{6} \leq \log \frac{n}{\delta} + \log \frac{\pi^2 k^2 \log^2(\frac{c}{\rho^2 + \|\theta\|_2^2/k})}{6} \quad (45)$$

$$\leq \log \frac{n}{\delta} + \log \frac{\pi^2 k^2 \log^2(\frac{c}{\rho^2})}{6} \leq \log \frac{n}{\delta} + \log \frac{\pi^2 k^2 \log^2(1 + \exp(4n/k))}{6} \quad (46)$$

$$\leq \log \frac{n}{\delta} + \log \frac{\pi^2 k^2 (2 + 4n/k)^2}{6} \leq \log \frac{n}{\delta} + 2 \log(6n + 3k) \quad (47)$$

By replacing the log term and utilizing  $\theta \in \hat{\Theta}$ , the absolute difference is bounded as follows:

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} [\mathcal{L}(\mathbb{D}; \theta + \epsilon) - \mathcal{L}(T; \theta + \epsilon)] \quad (48)$$

$$= \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} [|\mathcal{L}(\mathbb{D}; \theta + \epsilon) - \mathcal{L}(T; \theta + \epsilon)|] \leq \sqrt{\frac{\frac{1}{4}k \log(1 + \frac{\|\theta\|_2^2}{k\sigma^2}) + \frac{1}{4} + \log \frac{n}{\delta} + 2 \log(6n + 3k)}{n - 1}} \quad (49)$$

Utilizing the inequality in Eq (48), we replace the Eq (33) as follows:

$$\begin{aligned} & \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} [|\mathcal{L}(\mathbb{D}; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)|] \\ & \leq \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} [|\mathcal{L}(T; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)|] + \sqrt{\frac{\frac{1}{4}k \log(1 + \frac{\|\theta\|_2^2}{k\sigma^2}) + \frac{1}{4} + \log \frac{n}{\delta} + 2 \log(6n + 3k)}{n - 1}} \end{aligned} \quad (50)$$

Finally, we are to bound the expectation term in R.H.S with the  $\max_{\|\epsilon\|_2 \leq \rho} |\mathcal{L}(T; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)|$  by utilizing the results from Laurent and Massart (2000) as follows:

$$z \sim \mathcal{N}(0, \gamma I) \rightarrow \|z\|^2 \leq k\gamma \left(1 + \sqrt{\frac{\log n}{k}}\right)^2 \text{ with probability at least } 1 - \frac{1}{\sqrt{n}} \quad (51)$$

Here we denote  $\rho = k\gamma \left(1 + \sqrt{\frac{\log n}{k}}\right)^2$ . To provide the upper-bound of  $\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} [|\mathcal{L}(T; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)|]$ , we partition the  $\epsilon$  space into those with  $\|\epsilon\|_2 \leq \rho$  and  $\|\epsilon\|_2 > \rho$ . As  $\|\epsilon\|_2 \leq \rho$  with probability at least  $1 - \frac{1}{\sqrt{n}}$ , we have:

$$\begin{aligned} & \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} [|\mathcal{L}(T; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)|] \\ & \leq \left(1 - \frac{1}{\sqrt{n}}\right) \max_{\|\epsilon\|_2 \leq \rho} |\mathcal{L}(T; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)| + \frac{1}{\sqrt{n}} l_{max} \end{aligned} \quad (52)$$

$$\leq \max_{\|\epsilon\|_2 \leq \rho} |\mathcal{L}(T; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)| + \frac{1}{\sqrt{n}} l_{max} \quad (53)$$

Here,  $l_{max} = \max_{\|\epsilon\|_2 \leq \rho} |\mathcal{L}(T; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)|$ . By replacing original expectation term to  $\max_{\|\epsilon\|_2 \leq \rho} |\mathcal{L}(T; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)|$ , Eq (50) is derived as follows:

$$\begin{aligned} & \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} [|\mathcal{L}(\mathbb{D}; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)|] \\ & \leq \max_{\|\epsilon\|_2 \leq \rho} |\mathcal{L}(T; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)| + \frac{1}{\sqrt{n}} l_{max} + \sqrt{\frac{\frac{1}{4}k \log(1 + \frac{\|\theta\|_2^2}{k\sigma^2}) + \frac{1}{4} + \log \frac{n}{\delta} + 2 \log(6n + 3k)}{n - 1}} \end{aligned} \quad (54)$$

With the bounded  $\theta$  with  $k$  dimensions, The summation of second term and last term could be asymptotically described as  $\sqrt{\frac{O(k + \log \frac{n}{\delta})}{n - 1}}$ . With the replacement of last two terms to the corresponding asymptotical term, we re-arrange above equation as follows:

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)} [|\mathcal{L}(\mathbb{D}; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)|] \leq \max_{\|\epsilon\|_2 \leq \rho} |\mathcal{L}(T; \theta + \epsilon) - \mathcal{L}(S; \theta + \epsilon)| + \sqrt{\frac{O(k + \log \frac{n}{\delta})}{n - 1}} \quad (55)$$

By re-phrasing each term in Eq (55) into the shorter description, we conclude the proof as follows:

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)}[\mathcal{L}_{abs}(\mathbb{D}, S; \theta + \epsilon)] \leq \max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{abs}(T, S; \theta + \epsilon) + \sqrt{\frac{O(k + \log \frac{n}{\delta})}{n-1}} \quad (56)$$

### A.5 Proof of Corollary 3.5

We first refer the Corollary 3.5 here as follows:

**Corollary A.4.** *If  $\mathcal{L}_{abs}(T, S; \theta) \leq \mathcal{L}_{abs}(\mathbb{D}, S; \theta)$  for  $\theta \in \hat{\Theta}$ , with probability at least  $1 - \delta$  over the choice of the training set  $T$  with  $|T| = n$ , the following holds:*

$$\left( \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)}[\mathcal{L}_{abs}(\mathbb{D}, S; \theta + \epsilon)] - \mathcal{L}_{abs}(\mathbb{D}, S; \theta) \right) / \rho \leq \left( \max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{abs}(T, S; \theta + \epsilon) - \mathcal{L}_{abs}(T, S; \theta) \right) / \rho + \sqrt{\frac{O(k + \log \frac{n}{\delta})}{n-1}} \quad (57)$$

*Proof.* We first revisit the resulting equation by Theorem 3.4 as follows:

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)}[\mathcal{L}_{abs}(\mathbb{D}, S; \theta + \epsilon)] \leq \max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{abs}(T, S; \theta + \epsilon) + \sqrt{\frac{O(k + \log \frac{n}{\delta})}{n-1}} \quad (58)$$

As we assume that  $\mathcal{L}_{abs}(T, S; \theta) \leq \mathcal{L}_{abs}(\mathbb{D}, S; \theta)$  for  $\theta \in \hat{\Theta}$ , we can extend Eq (58) as follows:

$$\mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)}[\mathcal{L}_{abs}(\mathbb{D}, S; \theta + \epsilon)] - \mathcal{L}_{abs}(\mathbb{D}, S; \theta) \leq \max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{abs}(T, S; \theta + \epsilon) - \mathcal{L}_{abs}(T, S; \theta) + \sqrt{\frac{O(k + \log \frac{n}{\delta})}{n-1}} \quad (59)$$

By dividing both terms by  $\rho$  we finish the proof. As  $\rho$  is controllable hyper-parameter, which is usually set to value between 0.01 and 0.5, we do not reflect  $\rho$  on the asymptotical term.

$$\left( \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \rho)}[\mathcal{L}_{abs}(\mathbb{D}, S; \theta + \epsilon)] - \mathcal{L}_{abs}(\mathbb{D}, S; \theta) \right) / \rho \leq \left( \max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{abs}(T, S; \theta + \epsilon) - \mathcal{L}_{abs}(T, S; \theta) \right) / \rho + \sqrt{\frac{O(k + \log \frac{n}{\delta})}{n-1}} \quad (60)$$

## B Further Analyses of LCMat

### B.1 Analyses on $L_{abs}$

First, we recap our objective as follows:

$$\min_S \max_{\|\epsilon\|_2 \leq \rho} \frac{\mathcal{L}_{abs}(T, S; \theta + \epsilon) - \mathcal{L}_{abs}(T, S; \theta)}{\rho} \quad (61)$$

As stated in the main paper, optimization of Eq (61) will lead to 1) the minimization of  $\mathcal{L}_{abs}(T, S; \theta + \epsilon)$ ; and 2) the maximization of  $\mathcal{L}_{abs}(T, S; \theta)$ , respectively. The minimization of  $\mathcal{L}_{abs}(T, S; \theta + \epsilon)$  is profitable, which is also shown in Theorem 1. The maximization of  $\mathcal{L}_{abs}(T, S; \theta + \epsilon)$  is beneficial to some extent, in that it slightly regularizes the over-fitting of  $S$  to  $T$  based on the current parameter  $\theta$ . However, it could also lead to the under-fitting of  $S$  based on  $\theta$  if  $\mathcal{L}_{abs}(T, S; \theta)$  increases too much. In our practical implementation, the value of  $\mathcal{L}_{abs}(T, S; \theta)$  is bounded or regularized during the optimization.

**Selection-based methods** For selection-based methods, A subset  $S$  is constructed from  $T$  as  $S \subseteq T$ , where our current parameter  $\theta$  is assumed to be pre-trained on  $T$ . As the optimization of  $\theta$  based on  $T$  incurs  $\mathcal{L}(T; \theta)$  to be small, we assume that the increase of  $\mathcal{L}_{abs}(T, S; \theta)$  is induced by the large value of  $\mathcal{L}(S; \theta)$  than  $\mathcal{L}(T; \theta)$ . Having said that,  $\mathcal{L}_{abs}(T, S; \theta)$  gets the bound from the fixed state of  $\theta$  and  $T$  as follows:

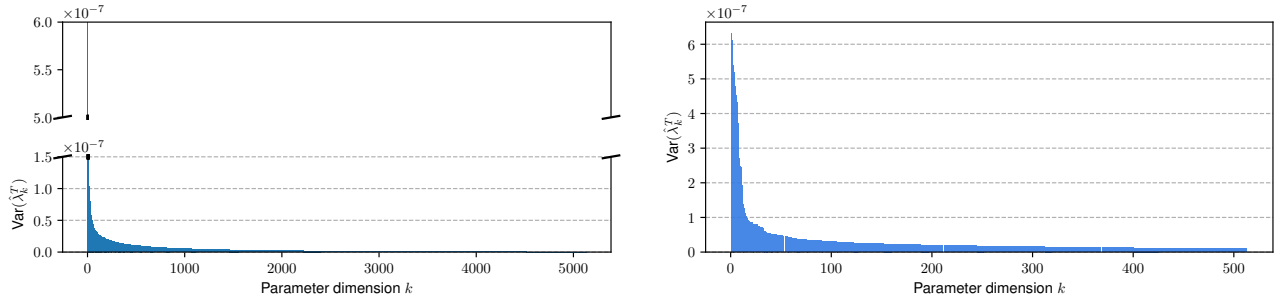
$$\mathcal{L}_{abs}(T, S; \theta) \leq \max_{S \subseteq T} \mathcal{L}_{abs}(T, S; \theta) = \max_{S \subseteq T} \left( \mathcal{L}(S; \theta) - \mathcal{L}(T; \theta) \right) \leq \max_{S \subseteq T} \mathcal{L}(S; \theta) - \mathcal{L}(T; \theta) \quad (62)$$

**Condensation-based methods** We recap our objective for application of condensation-based methods, LCMat-C, as follows:

$$\begin{aligned} \min_S \mathbb{E}_{\theta^0 \sim P_{\theta^0}} \left[ \sum_k \mathcal{D}(\bar{\mathbf{g}}_{\theta_k}^T, \bar{\mathbf{g}}_{\theta_k}^S) + \frac{1}{2} \rho |\text{Var}(\mathbf{G}_{\theta_k}^T) - \text{Var}(\mathbf{G}_{\theta_k}^S)| \right] \\ \text{s.t. } \theta_{t+1} = \theta_t - \eta \bar{\mathbf{g}}_{\theta_t}^T \text{ for } t = 0, \dots, k-1. \end{aligned} \quad (63)$$

As noted in the main paper, the objective is composed of 1)  $\mathcal{D}(\bar{\mathbf{g}}_{\theta_k}^T, \bar{\mathbf{g}}_{\theta_k}^S)$ , which is averaged gradient matching between  $T$  and  $S$ ; and 2)  $|\text{Var}(\mathbf{G}_{\theta_k}^T) - \text{Var}(\mathbf{G}_{\theta_k}^S)|$ , which is gradient variance matching between  $T$  and  $S$ . In practice, gradient variance matching is conducted based on the classifier parameters, where the classifier parameter weight and bias term is denoted as  $w$  and  $b$ , respectively. We utilize the findings from Rame et al. (2022) as follows:

When we utilize cross-entropy as a loss function, the derivative of sample  $(x, y)$  with respect to  $b$  is  $\nabla_b \ell(x, y; \theta) = (\hat{y} - y)$ , where  $\hat{y}$  is softmax output; and  $y$  is true label. Hence, when we compute the gradient variance based on a certain dataset  $D$ , the gradient variance is computed as  $\mathbf{v}_b^D = \frac{1}{|D|} \sum_{i=1}^{|D|} (\hat{y}_i - y_i)^2$ , which is equivalent to the mean squared error between the  $\hat{y}$  and  $y$ . Accordingly, the gradient variance matching of  $T$  and  $S$  based on the classifier bias term is equivalent to matching the mean squared error of  $T$  and  $S$ . Although the exact loss function is defined as cross-entropy, matching the mean squared error implicitly regularizes the difference between  $\mathcal{L}(T; \theta)$  and  $\mathcal{L}(S; \theta)$  to be small during the condensation procedure.



(a) Sorted  $\text{Var}(\hat{\lambda}_k^T)$  for whole dimensions of classifier parameter. (b) Sorted  $\text{Var}(\hat{\lambda}_k^T)$  for Top-500 dimensions of classifier parameter.

Figure 6: Empirically measured and sorted  $\text{Var}(\hat{\lambda}_k^T)$  for whole dimensions of classifier parameters in ResNet-18 on learning with CIFAR-10. As we construct the subset based on class-wise comparison, we measure  $\text{Var}(\hat{\lambda}_k^T)$  from the samples with same class. (Airplane for these figures)

## B.2 Analyses on Sub-Dimension Selection

a As noted in the main paper, we recap the sub-dimension selection criteria from whole parameter dimension. Let  $\mathcal{K}$  be a set of indexes for  $K$  sub-dimensions on  $\theta$ . We select  $K$  dominant sub-dimensions based on the variance of  $\hat{\lambda}_k^T = [\hat{\lambda}_{i,k}^T]_{i=1}^{|T|}$  for each  $k$ , which is denoted by the set  $\mathcal{K} = \underset{\mathcal{K}, |\mathcal{K}|=K}{\text{argmax}} \sum_{j \in \mathcal{K}} \text{Var}(\hat{\lambda}_k^T)$ . We assume that the large variance from specific parameter dimension means that there is a big difference in the corresponding eigenvalue of per-sample Hessian matrix for each sample. The difference between the averaged eigenvalue gap between the randomly selected subset and the entire training dataset would also likely to be large. In Figure 6,  $\text{Var}(\hat{\lambda}_k^T)$  shows the long-tailed distribution with concentration on specific dimensions from whole dimensions of parameter. In practice over the experiments of ResNet-18, we choose Top-100 dimensions from 5130 dimensions of classifier parameters. Although setting  $K = 100$  shows robust results over the experiments with ResNet-18, the optimal  $K$  could be slightly different if we change the network structure for measuring  $\text{Var}(\hat{\lambda}_k^T)$ .

## B.3 Discussion on the Limitations and Social Impacts of LCMat

**Limitations** The calculation of Hessian matrix over the model parameter induces the computational overhead during the optimization. As our method introduces the computation of Hessian over the classifier parameter, the computation of Hessian matrix could be costly when the number of feature dimensions and class dimensions further increases from the current experimental setting.

**Social Impacts** The data selection inevitably accompanies the discrimination of some samples than other samples, which are discarded from the dataset reduction procedure. Recently, Dong et al. (2022) found out that the condensation-based methods can be utilized to relieve the privacy issues by erasing the privacy-related information of each sample during the condensation. As we provide an application of our method for condensation-based method, we conjecture that our method can also be utilized as a privacy-robust method for dataset reduction task.

## C Technical Survey of Methods for Dataset Reduction

### C.1 Selection-based Methods

Selection-based methods find a data subset  $S \subset T$  that satisfies the cardinality constraint while maximizing the objective defined by the informativeness of  $S$ . We report details of the previous researches of selection-based methods in this section.

**Herding** (Welling, 2009) selects data points considering the distance between the feature center of the full dataset; and the feature center of the selected subset, and it selects samples to regularize the centers from each dataset to be similar.

**k-CenterGreedy** (Sener and Savarese, 2018) solves the coresets selection problem as k-Center problem (minimax facility location (Wolf, 2011)). Since k-Center problem is NP-Hard, it provides an approximate greedy solution for the problem by firstly selecting any sample as initialization and adding samples with maximum distances that have not been included to the coresets gradually.

**ContextualDiversity** (Agarwal et al., 2020) is similar to Sener and Savarese (2018), but it calculates the distance between two feature inputs using the summation of KL-divergence and reverse KL-divergence.

**Forgetting** (Toneva et al., 2018) assumes samples, which are not forgettable during the training procedure, are reducible. It defines forgetting as the event of wrong classification a sample when the model prediction of the sample was correct in the previous epoch. After a few epochs of training, samples are selected based on the number of forgotten times, which is counted for each sample. Therefore, it requires 1) saving all model prediction results from whole iterations and 2) an adequate number of training to get credible forgetting score.

**GraND** (Paul et al., 2021) calculates the expectation of the loss gradient with regard to model parameter. It is analytically regarded as the contribution of each sample to the averaged training loss. GraND also utilizes the outputs from multiple models, where each model is randomly initialized. Since these two methods both necessarily requires multiple times of model training with full dataset, we consider the framework of these methods is quiet different from our method. Hence, we do not report them as our baselines.

**Uncertainty** based methods (Coleman et al., 2019), which include LeastConfidence, Entropy and Margin in our baselines, assume that data samples with lower level of model prediction confidence would have larger impact on the construction of decision boundary. The scores of LeastConfidence, Entropy and Margin are defined as  $1 - \max_{i=1, \dots, C} P(\hat{y} = i|x)$ ,  $-\sum_{i=1}^C P(\hat{y} = i|x) \log P(\hat{y} = i|x)$ , and  $1 - \min_{y \neq \hat{y}} (P(\hat{y}|x) - P(y|x))$ , respectively. They select samples based on the computed scores in descending order.

Gradient-based methods minimize the distance between the gradients from the training dataset  $T$ ; and the (weighted) gradients from  $S$  as follows:

$$\min_{\mathbf{w}, S} \mathcal{D} \left( \frac{1}{|T|} \sum_{(x,y) \in T} \nabla_{\theta} \ell(x, y; \theta), \frac{1}{\|\mathbf{w}\|_1} \sum_{(x,y) \in S} w_x \nabla_{\theta} \ell(x, y; \theta) \right) \quad (64)$$

s.t.  $S \subset T, w_x \geq 0$

Here,  $\mathbf{w}$  is the vector of learnable weights for the data instances in  $S$ ;  $\|\mathbf{w}\|_1$  is 11-norm of  $\mathbf{w}$ ; and  $\mathcal{D}$  measures the distance between gradients.

To solve the problem, **Craig** (Mirzasoleiman et al., 2020) converts Eq (64) into the submodular maximization problem, and this research utilizes the greedy approach to optimize Eq (64).

Compared to Craig (Mirzasoleiman et al., 2020), **GradMatch** (Killamsetty et al., 2021a) utilizes orthogonal matching pursuit algorithm (Elenberg et al., 2018a) and squared  $L_2$  regularization term over  $\mathbf{w}$  to stabilize the optimization.

**Glistner** (Killamsetty et al., 2021b) introduces the generalization-based method, which results in the extraction of subsets which approximate the gradient of a training dataset or additional validation dataset well.



**AdaCore** (Pooladzandi et al., 2022) replaces  $\nabla_{\theta}l(x, y; \theta)$  in Eq (64) with a preconditioned gradient with the Hessian matrix, which leverages the second-order information for optimization. It firstly suggests a way of utilizing hessian information for coreset selection, however, the optimization is conducted to match the pre-conditioned gradients, which could also be generalized into loss-curvature matching method based on the pre-conditioned gradients.

## C.2 Condensation-based Methods

**DC** (Zhao et al., 2020) formulates the condensation method as a gradient matching task between the gradients of deep neural network weights, that are trained on the original and our synthetic data. Recently, Kim et al. (2022a) have introduced bag-of-tricks to improve the condensation quality with the gradient matching objective. It should be noted that these tricks could be orthogonally applied upon the choice of objective functions.

**DSA** (Zhao and Bilen, 2021a) proposes the Differentiable Siamese Augmentation (DSA), which utilizes the same data transformation to original data instances and synthetic data instances at each training iteration. Additionally, it enables the update of data transformation policy by back-propagating the gradient of the loss with respect to synthetic data into the augmentation parameters. Similar to Kim et al. (2022a), DSA are orthogonally applied upon the condensation objectives.

**DM** (Zhao and Bilen, 2021b) proposes matching feature distributions of the original dataset and synthetic dataset in sampled embedding spaces. As feature matching do not necessarily need the bi-level optimization between the model parameter,  $\theta$ , and the condensed dataset,  $S$ , it significantly reduces the computational costs of the gradient matching Zhao et al. (2020). However, the distribution matching do not provide the theoretical meaning of the introduced objective.

**KIP** (Nguyen et al., 2021) proposes a kernel-based objective which utilizes infinitely-wide neural networks. As condensed dataset is equivalent to the kernel inducing points from the kernel ridge-regression, it could be recognized as dataset summarization with kernel.

## D Experimental Details and Further Results

### D.1 Experimental Details

**Coreset Selection Evaluation** For coreset selection task, we use batch size of 128 for both CIFAR-10 dataset and CIFAR-100 dataset, for both model training for coreset selection; and model training with the selected instances. For the model optimization, We use SGD optimizer which utilizes learning rate of 0.1, momentum of 0.9, and weight decay (L2 regularization) parameter of  $5 \times 10^{-4}$ . After the extraction of  $S$ , we train the model with the selected instances,  $S$ , for 200 epochs. To validate the robustness of LCMat-S for each fraction budget, we report results with fraction over [0.1%, 0.5%, 1%, 5%, 10%, 20%, 30%]. We omit fraction of 0.1% condition for CIFAR-100, which chooses only one sample per class. We also report the performance of the model trained with the full dataset (100%). We consider it as the upper bound. For augmentation module, we utilize RandomCrop with reflection padding 4, RandomHorizontalFlip with probability 0.5, and Normalization for both CIFAR-10 and CIFAR-100 dataset. There are methods which require either outputs of a model or gradient signals from a model for each sample. To get such information, we trained a model with random initialized parameter for 10 epochs (Please refer to Appendix 4.5 for the sensitivity analysis on the number of this training epochs.). For gradient matching method such as Craig, GradMatch, Glister and AdaCore, we use the gradient signal of the last layer of the model because of the computation issue, as they did in the original paper (Mirzasoleiman et al., 2020; Killamsetty et al., 2021a,b; Pooladzandi et al., 2022). We select samples with class-balanced manner, meaning that the number of samples in the selected subset for each class should be balanced.

**Condensed Dataset Evaluation** As we learn  $\theta$  with  $T$  from the inner loop of bi-level optimization, we learn  $\theta$  with 1 epoch per one inner loop. The learning rates for model network and data variable are set to 0.01 and 0.005, respectively. Similar to Kim et al. (2022a), we utilize augmentation sequecne of color transform, crop, and cutout for data objective learning. Additionally, the initialization of the synthetic data is set to noise initialization. The evaluation scenario comes with the fraction budget, where we set 10, 50 samples per class as a practice setting.

### D.2 Wall-Clock Time Analyses of Selection-based Methods

In this section, we compare the computation time taken for each method over the different fraction budget: 0.1%, 1%, and 10%. Wall-clock time calculation includes 1) pre-training model with full dataset (with 10 epochs); and 2) subset selection process. As Uniform do not need the process of pre-training model with full-dataset, we skip the process for Uniform.

AdaCore and LCMat-S, which are methods which utilizes Hessian matrix during the selection, show significant increase of Wall-Clock time on the large fraction budget. It should be noted that the computation time of LCMat-S could be reduced if we utilize the faster approximation of  $L_1$  norm, which is introduced on the computation of Hessian matrix difference. In addition, it should be noted that AdaCore and LCMat-S only shows the consistently competitive performances over the Uniform baseline, which emphasizes the importance of modelling Hessian matrix during the selection procedure.

Fraction	Uniform	C-Div	Herding	k-Center	L-Conf	Entropy	Margin	Craig	GradMatch	Glister	AdaCore	LCMat-S
0.001		201.15	205.09	458.78	202.35	203.93	204.14	224.82	199.39	204.68	204.38	458.92
0.01	0.04	205.88	202.69	455.34	204.32	201.68	204.27	234.17	209.72	203.56	304.50	627.50
0.1		202.89	205.86	458.80	203.68	196.51	202.23	246.97	259.59	204.74	1255.49	1400.56

### D.3 Results with Inception-v3

Here, we report the test accuracy of the Inception-v3 trained by the  $S$  from each method. As reported in the main paper, we evaluate  $S$  with different fractions in dataset reduction, which is the cardinality budget of  $S$  from  $T$ . Similar to the results computed from the ResNet-18 network, Uniform shows competitive performances over other baselines, meaning the weak robustness of the existing selection methods. LCMat-S shows competitive performances over the implemented baselines by relieving the over-fitting issue of  $S$  to the provided  $\theta$ . Comparing the results from ResNet-18 and Inception-V3, the results of LCMat-S from Inception-V3 shows degraded performance than the ones from ResNet-18. As the number of classifier dimensions from Inception-V3 is 4 times bigger than the one from ResNet-18, our method could not cover the whole dimensions to compute the corresponding Hessian matrix. We assume that sub-dimension computation of Hessian matrix could be naive when original parameter dimension is too large to cover the whole dimensions by sub-dimension computation.

Table 6: Coreset selection performances on CIFAR10. We train randomly initialized Inception-v3 on the coresets selected by different methods and then test on the real testing set.

Fraction	CIFAR10							
	0.1%	0.5%	1%	5%	10%	20%	30%	100%
Uniform	17.59±2.9	27.24±2.3	35.29±0.3	60.09±1.1	76.73±1.3	85.52±0.6	89.25±0.5	95.62±0.1
C-Div	11.94±0.4	19.26±0.7	21.9±2.4	35.89±3.7	55.18±2.4	82.99±0.7	90.3±0.5	
Herding	14.52±0.5	26.03±2.4	32.06±2.4	49.86±4.4	64.98±1.5	75.56±1.0	80.99±0.2	
k-Center	15.81±1.0	20.4±0.4	25.48±0.4	48.8±3.3	75.47±1.8	85.72±0.4	90.08±0.1	
L-Conf	13.36±1.3	14.88±0.7	19.19±2.2	34.85±3.1	60.75±2.8	82.66±1.3	89.92±0.1	
Entropy	12.73±0.6	16.26±2.2	17.91±1.5	37.53±3.1	54.94±1.8	82.54±1.1	89.97±0.9	
Margin	15.29±1.3	23.81±1.5	26.71±1.2	43.14±1.5	63.29±3.3	83.36±1.2	90.14±0.5	
Craig	13.54±0.8	22.50±1.8	24.55±5.9	38.05±1.6	52.13±6.6	71.00±3.1	82.68±1.7	
GradMatch	12.73±1.1	18.24±1.4	18.69±0.6	35.56±2.9	50.91±4.5	68.95±1.3	83.34±0.6	
Glister	15.52±1.5	21.82±1.6	22.11±1.9	34.71±0.7	48.98±4.6	70.13±2.9	84.33±0.6	
AdaCore	15.79±1.8	27.48±1.2	33.93±0.2	58.57±1.5	71.97±3.9	86.2±0.6	90.54±0.4	
LCMat-S	18.55±1.8	29.33±0.5	36.09±0.6	53.23±2.1	69.14±1.3	85.21±1.2	89.89±0.2	

### D.4 Selected Images for All Class in Cifar-10

Figure 7 is the visualization of selected samples for CIFAR-10 dataset under ResNet-18 network structure. Whole selected images were displayed without any cherry-picking. LCMat-S selects a set of examples with diverse characteristics, e.g. the diverse shape of each object, different backgrounds without redundancy.

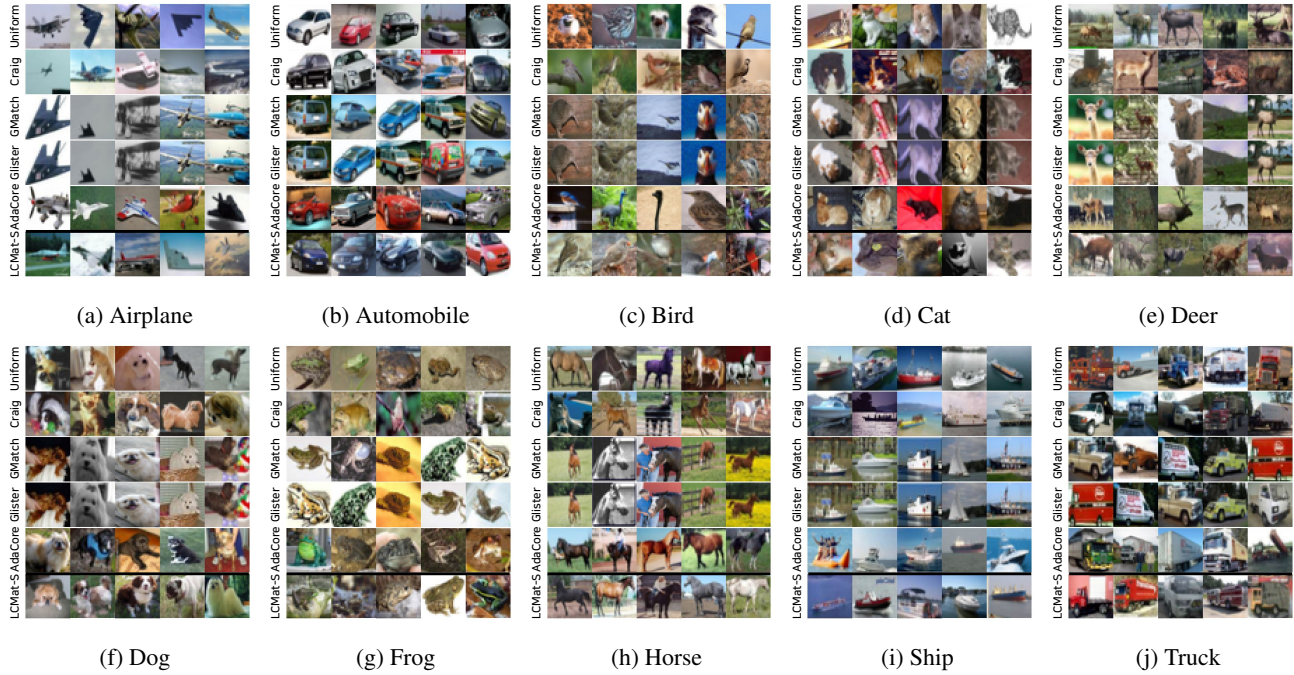


Figure 7: A set of images selected from each method. All samples are selected in a class-balanced way. We report selected images for 0.1% fraction here (Total of 50 images, 5 images per class).