

A Appendix

A.1 Related work

Coreset methods To facilitate the training of deep learning models, various methods have been proposed to select informative or representative samples based on the uncertainty of models toward these samples such as Sachdeva et al. (2021) and Coleman et al. (2020). Another line of work selects samples based on their loss difference or degree of error. Methods such as Forgetting Events Toneva et al. (2019), GraNd Paul et al. (2023) have employed this strategy to prioritize samples. These methods aim to reduce the variance of the gradient, thereby improving efficiency of reducing the loss.

Additionally, some methods emphasize the centrality of features or embeddings, forming subsets that best represent clusters of samples. Examples include Herding Chen et al. (2012), K-Center Greedy Ding et al. (2019), and Prototypes Sorscher et al. (2023). Another category of methods bases their selection strategy on observations from a validation set Killamsetty et al. (2021b), leveraging additional information to identify the samples most beneficial for training.

Despite lacking rigorous theoretical proof for the advantages of selection strategies, these approaches have demonstrated empirical improvements in speed or performance. Gradient-based methods Mirzasoleiman et al. (2020); Killamsetty et al. (2021a); Pooladzandi et al. (2022) on the other hand, select samples that best approximate the gradient of the entire dataset (training or validation sets). These methods offer theoretically sound support and provide guarantees for the training process. Additionally, several works Pooladzandi et al. (2022); Shin et al. (2023) in this direction propose to further match the subset with the loss landscape using information in Hessian matrix and show better convergent result and generalization performance. Our method builds on this direction and addresses the shortcomings of previous work.

Smoothness and Stability The optimization of deep learning models has been active area of studied to understand the reasons behind their superior performance in practice. For instance, the Random Weight Perturbation (RWP) algorithm has been shown to smooth the objective function (Bisla et al., 2022; Li et al., 2024) and improve generalization error (Zhou et al., 2019; Jin et al., 2019; Wang & Mao, 2022) despite its simplicity. By perturbing model weights, several studies have demonstrated an increased ability to escape minima with poor generalization and enhance the stability of the optimization process. Our work draws connection to this approach. Instead of explicitly optimizing using a smoothed objective, we select samples that capture similar features, thereby achieving comparable effects. This novel perspective enables us to leverage the benefits of smoothness without directly modifying the optimization objective.

Bayesian Methods in Deep Learning Various parameter distributions have been proposed in deep learning to address tasks such as uncertainty estimation, out-of-distribution detection, and classification. Markov chain Monte Carlo (MCMC) methods Chen et al. (2014); Welling & Teh (2011) leverage gradient information for inference, while Laplace approximation-based approaches MacKay (1992); Kirkpatrick et al. (2017); Ritter et al. (2018a) employ Gaussian distributions with the Fisher information matrix or Hessian as the covariance matrix. Other methods explore different strategies: Maddox et al. (2019) average models across different time steps, and Fort et al. (2020) utilize models trained independently with different random seeds. In our work, we evaluate models derived from various posteriors to analyze the trade-offs associated with different sampling strategies.

A.2 Second order proof

Theorem A.1. Suppose a subset $S' \subset S$ is $\{\sigma, \epsilon, \bar{w}\}$ -stable and let the Hessian difference be $H_{S', \bar{w}} - H_{S, \bar{w}} =: \mathcal{E}$, and model has d parameters then,

(1) The gradient at difference of subset at w is upper bounded

$$|\nabla l_S(\bar{w}) - \nabla l_{S'}(\bar{w})| \leq \frac{1}{2}(c_1\sigma^2d + \sqrt{c_1^2\sigma^4d^2 + 4\epsilon}) = \mathcal{O}(\epsilon^{\frac{1}{2}}) \quad (10)$$

(1) The Hessian difference matrix \mathcal{E} satisfies:

$$\|\mathcal{E}\| \leq c_1\sigma d + \frac{1}{\sigma}\sqrt{c_1^2\sigma^4d^2 + \sigma(\epsilon - c_2^2\sigma^2d)} \quad \text{and} \quad \text{tr}(\mathcal{E}^2) \leq \frac{\epsilon - c_2^2\sigma^2d}{\sigma(1 - 2c_1\sigma d)} \quad (11)$$

868 (2) The difference between newton step of two subset is bounded. $(\lambda_{\max}, \lambda_{\min})$ are largest and
 869 smallest eigenvalue in H_{S,w^*}

$$\begin{aligned} |H_{S',\bar{w}}^{-1} \nabla l_{S'}(\bar{w}) - H_{S,\bar{w}}^{-1} \nabla l_S(\bar{w})| &\leq \frac{1}{\lambda_{\min}} \frac{1}{2} (c_1 \sigma^2 d + \sqrt{c_1^2 \sigma^4 d^2 + 4\epsilon}) + c \frac{\lambda_{\max}^{\frac{1}{2}}}{\lambda_{\min}^2} (c_1 \sigma d + \\ &\quad \frac{1}{\sigma} \sqrt{c_1^2 \sigma^4 d^2 + \sigma(\epsilon - c_2^2 \sigma^2 d)}) + \frac{\lambda_{\max}^{\frac{1}{2}}}{\lambda_{\min}^2} \frac{1}{2} (c_1 \sigma^2 d + \sqrt{c_1^2 \sigma^4 d^2 + 4\epsilon}) (c_1 \sigma d \\ &\quad + \frac{1}{\sigma} \sqrt{c_1^2 \sigma^4 d^2 + \sigma(\epsilon - c_2^2 \sigma^2 d)}) + \mathcal{O}(\|\mathcal{E}\|^2) \end{aligned}$$

870 *Proof.* Let $f(\bar{w}) = \nabla l_{S'}(\bar{w}) - \nabla l_S(\bar{w})$ and $\nabla f(\bar{w}) = \nabla^2 l_{S'}(\bar{w}) - \nabla^2 l_S(\bar{w})$ (difference in terms
 871 of Hessian)

872 **Assumption 1** Suppose we have the $\nabla f(\bar{w})$ being $2c_1$ -Lipschitz Hessian i.e.,

$$\|\nabla f(w) - \nabla f(\bar{w})\| \leq 2c_1 |w - \bar{w}| \quad (12)$$

873 **Assumption 2** Suppose we have the $\nabla f(\bar{w})$ being bounded below and above

$$2c_2 I \preceq \nabla f(w), \quad \forall w. \quad (13)$$

874 **Assumption 3** (Symmetric and non-singular) $\nabla f(w)$ is symmetric and non-singular.

875 With assumption 1 and assumption 2, we can bound the $f(\bar{w})$ by the following through Hefferon
 876 (2017):

$$c_2 |w - \bar{w}| \leq |f(w) - f(\bar{w}) - \nabla f(\bar{w})(w - \bar{w})| \leq c_1 |w - \bar{w}| \quad \forall w, \bar{w} \quad (14)$$

877 The assumption is aiming to capture the degree of change of function using polynomial terms. For
 878 more discuss about the assumption of the theory, please refer to appendix D

879 we start with the following

$$\begin{aligned} \epsilon &\geq \int f(w)^2 p(w) dw \\ &= \int (f(\bar{w}) + f(w) - f(\bar{w}))^2 p(w) dw \\ &= f(\bar{w})^2 + 2f(w^*) \int (f(w) - f(\bar{w})) dw + \int (f(w) - f(\bar{w}))^2 p(w) dw \\ &\geq |f(\bar{w})|^2 - c_1 \sigma^2 d |f(\bar{w})| + \int (f(w) - f(\bar{w}))^2 p(w) dw \end{aligned} \quad (15)$$

880 We first focus on the first two terms

$$\epsilon \geq |f(\bar{w})|^2 - c_1 \sigma^2 d |f(\bar{w})| \quad (16)$$

881 By solving the equation, we can obtain upper bound on the $|f(\bar{w})|$ as following:

$$|f(\bar{w})| \leq \frac{1}{2} (c_1 \sigma^2 d + \sqrt{c_1^2 \sigma^4 d^2 + 4\epsilon}) = \mathcal{O}(\epsilon^{\frac{1}{2}}) \quad (17)$$

882 as a check, if we use linear approximation (i.e., $c_1 = 0$) we will have

$$|f(\bar{w})| \leq \sqrt{\epsilon} \quad (18)$$

883 Now, obtain upper bound on the gradient difference at \bar{w} , we continue on the cross terms

$$\begin{aligned}
\epsilon &\geq \int (f(w) - f(\bar{w}))^2 p(w) dw \\
&= \int (f(w) - f(\bar{w}) - \nabla f(\bar{w})(w - \bar{w}) + \nabla f(\bar{w})(w - \bar{w}))^2 p(w) dw \\
&= \int (f(w) - f(\bar{w}) - \nabla f(\bar{w})(w - \bar{w}))^2 p(w) dw \\
&+ 2 \int (f(w) - f(\bar{w}) - \nabla f(\bar{w})(w - \bar{w})) \nabla f(\bar{w})(w - \bar{w}) p(w) dw + \int (w - \bar{w}) \nabla f(\bar{w})^2 (w - \bar{w}) p(w) dw
\end{aligned} \tag{19}$$

884 we continue by noticing that first term is lower bounded in our assumption.

$$\begin{aligned}
\epsilon &\geq c_2^2 \sigma^2 d \\
&+ 2 \int (f(w) - f(\bar{w}) - \nabla f(\bar{w})(w - \bar{w})) \nabla f(\bar{w})(w - \bar{w}) p(w) dw + \int (w - \bar{w}) \nabla f(\bar{w})^2 (w - \bar{w}) p(w) dw
\end{aligned} \tag{20}$$

885 Now address the cross term,

$$\begin{aligned}
2 \int (f(w) - f(\bar{w}) - \nabla f(\bar{w})(w - \bar{w})) \nabla f(\bar{w})(w - \bar{w}) p(w) dw &\geq -2c_1 \int |w - \bar{w}| \mathcal{E}(w - \bar{w}) |p(w) dw \\
&\geq -2c_1 \max_i \lambda_{\mathcal{E},i} \int (w - \bar{w})^2 p(w) dw \\
&\geq -2c_1 \sigma^2 d \max_i \lambda_{\mathcal{E},i}
\end{aligned} \tag{21}$$

886 Now, we address the last term

$$\int (w - \bar{w}) \mathcal{E}^2(w - \bar{w}) 2\pi^{-\frac{d}{2}} \det(\sigma I)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(w - \bar{w})(\sigma I)^{-1}(w - \bar{w})\right) dw \tag{22}$$

887 By change of variable $u = \nabla f(\bar{w})(w - \bar{w})$, we can obtain the following by assuming that \mathcal{E} is not
888 degenerate:

$$\begin{aligned}
&\int |\det(\mathcal{E}^{-1})| u^T u 2\pi^{-\frac{d}{2}} \det(\sigma I)^{-\frac{1}{2}} \exp\left(-\frac{1}{2} u^T \mathcal{E}_1^{-1} (\sigma I)^{-1} \mathcal{E}^{-1} u\right) du \\
&= |\det(\mathcal{E}^{-1})| \frac{\det(\mathcal{E}(\sigma I) \mathcal{E})^{\frac{1}{2}}}{\det((\sigma I))^{\frac{1}{2}}} \text{tr}(\mathcal{E}(\sigma I) \mathcal{E}) \\
&= \text{tr}(\sigma I \mathcal{E}^2) \\
&\geq \sigma \text{tr}(\mathcal{E}^2) \\
&\geq \sigma \max_i \lambda_{\mathcal{E},i}^2
\end{aligned} \tag{23}$$

889 We assume the H^{-1} and $\nabla f(\bar{w})$ are symmetry matrix. Here, we utilize two identities for the first
890 equility and second inequility:

$$\text{tr}(ABC) = \text{tr}(BCA) \tag{24}$$

891 The second identity follows the following derivation by first noting that symmetry matrix can be
892 decompose into the spectral form $A = \sum_i \lambda_{A,i} v_i v_i^T$

$$\begin{aligned}
\text{tr}(AB) &= \sum_i \lambda_{B,i} (v_i^T A v_i) \\
&\geq \min_i \lambda_{B,i} \sum_i (v_i^T A v_i) \\
&= \min_i \lambda_{B,i} \text{tr}(A)
\end{aligned} \tag{25}$$

Now, we put everything together,

$$\epsilon \geq c_2^2 \sigma^2 d - 2c_1 \sigma^2 d \max_i \lambda_{\mathcal{E},i} + \sigma \max_i \lambda_{\mathcal{E},i}^2 \tag{26}$$

we can solve for the largest difference in eigenvalue

$$c_1 \sigma d + \frac{1}{\sigma} \sqrt{c_1^2 \sigma^4 d^2 + \sigma(\epsilon - c_2^2 \sigma^2 d)} \geq \max_i \lambda_{\mathcal{E},i} \tag{27}$$

Therefore, we can have that

$$|H_{S',\bar{w}} - H_{S,\bar{w}}| \leq \mathcal{O}(\epsilon^{\frac{1}{2}}) \tag{28}$$

We can also obtain upper bound on the difference in Hessian in terms of trace

$$\begin{aligned}
\epsilon - c_2^2 \sigma^2 d &\geq -2c_1 \sigma^2 d \max_i \lambda_{\mathcal{E},i} + \sigma \max_i \lambda_{\mathcal{E},i}^2 \\
&\geq -2c_1 \sigma^2 d \text{tr}(\mathcal{E}^2) + \sigma \text{tr}(\mathcal{E}^2) \\
&\geq \sigma(1 - 2c_1 \sigma d) \text{tr}(\mathcal{E}^2)
\end{aligned} \tag{29}$$

and therefore

$$\frac{\epsilon - c_2^2 \sigma^2 d}{\sigma(1 - 2c_1 \sigma d)} \geq \text{tr}(\mathcal{E}^2) \tag{30}$$

Now, to prove the newton step is also similar, we write the overall into the following equation

$$|H_{S',\bar{w}}^{-1} \nabla' l(\bar{w}) - H_{S,\bar{w}}^{-1} \nabla l_S(\bar{w})| = |(H_{S,\bar{w}} + \mathcal{E})^{-1} (\nabla l_S(\bar{w}) + \mathcal{E}_2) - H_{S,\bar{w}}^{-1} \nabla l_S(\bar{w})| \tag{31}$$

To obtain the upper bound of the Hessian inverse, we use inverse perturbation theory

$$(H_{S,\bar{w}} + \mathcal{E})^{-1} = H_{S,\bar{w}}^{-1} - H_{S,\bar{w}}^{-1} \mathcal{E} H_{S,\bar{w}}^{-1} + \mathcal{O}(\|\mathcal{E}\|^2) \tag{32}$$

Now, we obtain upper bound on the magnitude of eigenvalue of the difference on the Hessian

for the subset. (i.e., $\max_i |\lambda_{\mathcal{E},i}| \leq c_1 \sigma d + \frac{1}{\sigma} \sqrt{c_1^2 \sigma^4 d^2 + \sigma(\epsilon - c_2^2 \sigma^2 d)}$ and the gradient difference

$|\nabla l_{s'}(w) - \nabla l_s(w)| \leq \frac{1}{2}(c_1 \sigma^2 d + \sqrt{c_1^2 \sigma^4 d^2 + 4\epsilon})$. We can follow to upper bound the desired

quantity $|H_{S',\bar{w}}^{-1} \nabla l_{S'}(w) - H_{S,\bar{w}}^{-1} \nabla l_s(w)|_2^2$. At this point, we assume that the gradient has bounded

magnitude $|\nabla l_S(w)| \leq c, \forall S, w$

Put in the above, and assume that the gradient has bounded magnitude $|\nabla l_S(w)| \leq c \forall S, w$

$$\begin{aligned}
&|H_{S',\bar{w}}^{-1} \nabla' l(\bar{w}) - H_{S,\bar{w}}^{-1} \nabla l_S(\bar{w})| \\
&= |(H_{S,\bar{w}} + \mathcal{E})^{-1} (\nabla l_S(\bar{w}) + \mathcal{E}_2) - H_{S,\bar{w}}^{-1} \nabla l_S(\bar{w})| \\
&\leq |H_{S,\bar{w}}^{-1} \mathcal{E}_2 - H_{S,\bar{w}}^{-1} \mathcal{E} H_{S,\bar{w}}^{-1} \nabla l_S(\bar{w}) + H_{S,\bar{w}}^{-1} \mathcal{E} H_{S,\bar{w}}^{-1} \mathcal{E}_2| + \mathcal{O}(\|\mathcal{E}\|^2) \\
&\leq |H_{S,\bar{w}}^{-1} \mathcal{E}_2| + |H_{S,\bar{w}}^{-1} \mathcal{E} H_{S,\bar{w}}^{-1} \nabla l_S(\bar{w})| + |H_{S,\bar{w}}^{-1} \mathcal{E} H_{S,\bar{w}}^{-1} \mathcal{E}_2| + \mathcal{O}(\|\mathcal{E}\|^2) \\
&\leq \frac{1}{\lambda_{\min}} \frac{1}{2} (c_1 \sigma^2 d + \sqrt{c_1^2 \sigma^4 d^2 + 4\epsilon}) + c \frac{\lambda_{\max}^{\frac{1}{2}}}{\lambda_{\min}^2} (c_1 \sigma d + \frac{1}{\sigma} \sqrt{c_1^2 \sigma^4 d^2 + \sigma(\epsilon - c_2^2 \sigma^2 d)}) + \frac{\lambda_{\max}^{\frac{1}{2}}}{\lambda_{\min}^2} \frac{1}{2} (c_1 \sigma^2 d + \sqrt{c_1^2 \sigma^4 d^2 + 4\epsilon}) (c_1 \sigma d + \frac{1}{\sigma} \sqrt{c_1^2 \sigma^4 d^2 + \sigma(\epsilon - c_2^2 \sigma^2 d)}) + \mathcal{O}(\|\mathcal{E}\|^2) \\
&\leq \mathcal{O}(\epsilon^{\frac{1}{2}})
\end{aligned} \tag{33}$$

At this point, we can conclude that

$$|H_{S',\bar{w}}^{-1}\nabla'_s l(\bar{w}) - H_{S,\bar{w}}^{-1}\nabla l_s(\bar{w})|_2^2 \leq \mathcal{O}(\epsilon) \quad (34)$$

□

A.3 Proof for Theorem 3.3

Next, the assumptions required for the proof are listed below:

Assumption 1. Bounded variance and unbiased estimator of the random sampling and coreset selection subroutine:

$$E[||\nabla l_S(w) - \nabla l(w)||^2] \leq \sigma_1^2 \quad (35)$$

$$E[\nabla l_S(w)] = \nabla l(w) \quad (36)$$

Assumption 2. α -lipschitz continuity:

$$l(w) - l(v) \leq \alpha ||w - v||_2 \quad (37)$$

Assumption 3. β -smoothness:

$$||\nabla l(w) - \nabla l(v)|| \leq \beta ||w - v||_2 \quad (38)$$

Assumption 4. $\epsilon_i \in \mathbb{R}^d$ sampled i.i.d from diagonal gaussian:

$$\epsilon_i \sim N(0, \sigma_2 I), \quad \forall i = 1 \dots M \quad (39)$$

There exist two randomnesses in our formulation. One is the subset obtained through random sampling from the whole dataset. The other randomness results from the noise inject to model weight.

$$\begin{aligned} l(w_{t+1}) &\leq l(w_t) + \nabla l(w_t)^T (w_{t+1} - w_t) + \frac{\beta}{2} ||w_{t+1} - w_t||^2 \\ &\leq l(w_t) - \eta_t \nabla l(w_t)^T \frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i) + \frac{\beta \eta_t^2}{2} ||\frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i)||^2 \end{aligned} \quad (40)$$

We rewrite the last term as follows:

$$\frac{\beta \eta_t^2}{2} ||\frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i)||^2 = \frac{\beta \eta_t^2}{2} (||\frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i) - \nabla l(w_t)||^2 + \frac{2}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i) \nabla l(w_t) - ||\nabla l(w_t)||^2) \quad (41)$$

Input the above into the original formulation:

$$\begin{aligned} l(w_{t+1}) &\leq l(w_t) - \eta_t \nabla l(w_t)^T \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i) + \frac{\beta \eta_t^2}{2} ||\frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i)||^2 \\ &\leq l(w_t) - \eta_t (1 - \eta_t \beta) \nabla l(w_t)^T \frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i) + \frac{\beta \eta_t^2}{2} (||\frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i) - \nabla l(w_t)||^2 - ||\nabla l(w_t)||^2) \\ &\leq l(w_t) - \eta_t (1 - \eta_t \beta) \nabla l(w_t)^T \frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i) - \frac{\beta \eta_t^2}{2} ||\nabla l(w_t)||^2 + \\ &\quad \beta \eta_t^2 ||\frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i) - \nabla l(w_t + \epsilon_i)||^2 + \beta \eta_t^2 ||\frac{1}{M} \sum_{i=1}^M \nabla l(w_t + \epsilon_i) - \nabla l(w_t)||^2 \end{aligned} \quad (42)$$

920 The last two terms are achieved through the identity $\|a - b\| \leq 2(\|a - c\| + \|b - c\|)$. We now want
 921 to resolve the second term in the formulation.

$$\begin{aligned}
 \nabla l(w_t)^T \frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i) &= \nabla l(w_t)^T \left(\frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i) - \nabla l(w_t) + \nabla l(w_t) \right) \\
 &= \|\nabla l(w_t)\|^2 + \nabla l(w_t)^T \left(\frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i) - \nabla l(w_t) \right) \\
 &= \|\nabla l(w_t)\|^2 + \\
 &\quad \nabla l(w_t)^T \left(\frac{1}{M} \sum_{i=1}^M (\nabla_S l(w_t + \epsilon_i) - \nabla l(w_t + \epsilon_i)) + \frac{1}{M} \sum_{i=1}^M \nabla l(w_t + \epsilon_i) - \nabla l(w_t) \right)
 \end{aligned} \tag{43}$$

922 The term can be simplified by taking the expectation over the randomness in a stochastic subset.

$$\begin{aligned}
 E_S[\nabla l(w_t)^T \frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i)] &= \|\nabla l(w_t)\|^2 + \nabla l(w_t)^T \left(\frac{1}{M} \sum_{i=1}^M \nabla l(w_t + \epsilon_i) - \nabla l(w_t) \right) \\
 &\leq \|\nabla l(w_t)\|^2 + \frac{1}{2} \|\nabla l(w_t)\|^2 + \frac{1}{2} \left(\left\| \frac{1}{M} \sum_{i=1}^M \nabla l(w_t + \epsilon_i) - \nabla l(w_t) \right\|^2 \right)
 \end{aligned} \tag{44}$$

923 The last term is achieved through the Cauchy-Schwarz inequality. Therefore,

$$-E_S[\nabla l(w_t)^T \frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i)] \leq -\frac{1}{2} \|\nabla l(w_t)\|^2 + \frac{1}{2} \left(\left\| \frac{1}{M} \sum_{i=1}^M \nabla l(w_t + \epsilon_i) - \nabla l(w_t) \right\|^2 \right) \tag{45}$$

924 We further simplified the formulation by taking the expectation over the randomness in noise injected
 925 to the model weight

$$-E_{S, \epsilon_i, i=1 \dots d}[\nabla l(w_t)^T \frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i)] \leq -\frac{1}{2} \|\nabla l(w_t)\|^2 + \frac{1}{2} \beta^2 \sigma_2^2 d \tag{46}$$

926 Next, we first solve the fifth term in the original formulation taking the expectation with respect to ϵ_i
 927 for each i .

$$\begin{aligned}
 \beta \eta_t^2 E[\left\| \frac{1}{M} \sum_{i=1}^M \nabla l(w_t + \epsilon_i) - \nabla l(w_t) \right\|^2] &\leq \beta \eta_t^2 E[\left\| \frac{1}{M} \sum_{i=1}^M \beta \epsilon_i \right\|^2] \\
 &\leq \beta^3 \eta_t^2 \frac{1}{M} \sum_{i=1}^M E[\|\epsilon_i\|^2] \\
 &\leq \beta^3 \eta_t^2 \sigma_2^2 d
 \end{aligned} \tag{47}$$

928 Finally, we deal with the term.

$$\beta \eta_t^2 \left\| \frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i) - \nabla l(w_t + \epsilon_i) \right\|^2 \tag{48}$$

929 Here, we assume two different errors that can arise in practice. The first is the random sampling
 930 batch created by sampling from the entire dataset. The second error originates from the coresot
 931 approximation. We formulate as follows:

$$\nabla l(w) = \nabla_S l(w) + \xi_1 + \xi_2 \quad (49)$$

932 The ξ_1 is the result of the stochasticity of the random batch generation. The ξ_2 is the error that
 933 originates from the selection of the coreset. We consider two different forms of error in ξ_2 . One is the
 934 absolute error and the other is that the error is propotional to the gradient. We formulate as follows:

$$E[||\xi_2||] \leq \epsilon \quad \text{or} \quad E[||\xi_2||] \leq \epsilon ||\nabla l(w)||, \quad \epsilon > 0 \quad (50)$$

935 Note: we assume here that these two errors ξ_1, ξ_2 are independent to each other.

936 **Situation 1** We first consider the case where the error is absolute.

$$\begin{aligned} \beta \eta_t^2 E_{S, \epsilon_i, i=1 \dots M} E \left[\left| \frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i) - \nabla l(w_t + \epsilon_i) \right|^2 \right] &\leq \beta \eta_t^2 \frac{1}{M^2} \sum_{i=1}^M E \left[\left| \nabla_S l(w_t + \epsilon_i) - \nabla l(w_t + \epsilon_i) \right|^2 \right] \\ &= \beta \eta_t^2 \frac{1}{M^2} \sum_{i=1}^M E \left[||\xi_{1,i}||^2 + 2\xi_{1,i}\xi_{2,i} + ||\xi_{2,i}||^2 \right] \\ &= \beta \eta_t^2 \frac{1}{M} (||\xi_1||^2 + 2\xi_1\xi_2 + ||\xi_2||^2) \\ &\leq \beta \eta_t^2 \frac{1}{M} \left(\frac{\sigma_1^2}{R} + \epsilon^2 \right) \end{aligned} \quad (51)$$

937 The R is the batch size for each batch of random sampling. The cross terms are eliminated due to the
 938 assumption 5.

939 Integrate those terms into the original formulation.

$$l(w_{t+1}) \leq l(w_t) - \eta_t ||\nabla l(w_t)||^2 + \frac{\eta_t(1 - \eta_t\beta)}{2} \beta^2 \sigma_2^2 d + \frac{\beta \eta_t^2 \sigma_1^2}{MR} + \frac{\beta \eta_t^2 \epsilon^2}{M} + \beta^3 \eta_t^2 \sigma_2^2 d \quad (52)$$

940 Here, we pick $\eta_t = \eta$ (fixed step size) and $\eta \leq \frac{1}{\beta}$. Rearrange and sum over time step, and we will
 941 have as follows:

$$\eta \sum_{t=0}^{T-1} ||\nabla l(w_t)||^2 \leq l(w_0) - l(w^*) + \frac{\beta^2 \sigma_2^2 d}{2} \sum_{t=0}^{T-1} \eta(1 + \eta\beta) + \frac{\beta \epsilon^2}{M} \sum_{t=0}^{T-1} \eta^2 + \frac{\beta \sigma_1^2}{MR} \sum_{t=0}^{T-1} \eta^2 \quad (53)$$

942 Here, we divide on both sides by $T\eta$

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} ||\nabla l(w_t)||^2 &\leq \frac{1}{T\eta} (l(w_0) - l(w^*)) + \frac{\beta^2 \sigma_2^2 d}{2T} \sum_{t=0}^{T-1} (1 + \eta\beta) + \frac{\beta \epsilon^2}{M} \eta + \frac{\beta \sigma_1^2}{MR} \eta \\ &= \frac{1}{T\eta} (l(w_0) - l(w^*)) + \frac{\beta^2 \sigma_2^2 d}{2} + \frac{\beta^3 \sigma_2^2 d}{2} \eta + \frac{\beta \epsilon^2}{M} \eta + \frac{\beta \sigma_1^2}{MR} \eta \end{aligned} \quad (54)$$

943 As we have control for both η and $\sigma_2^2 d$, we pick $\sigma_2^2 d = \frac{1}{M\sqrt{T}}$ and $\eta = \min\{\frac{1}{\sqrt{T}}, \frac{1}{\beta}\}$ and, therefore,

$$\frac{1}{T} \sum_{t=0}^{T-1} ||\nabla l(w_t)||^2 \leq \frac{1}{T} (l(w_0) - l(w^*)) \max\{\sqrt{T}, \beta\} + \frac{1}{\sqrt{T}} \left(\frac{\beta^2}{2M} + \frac{\beta \epsilon^2}{M} + \frac{\beta \sigma_1^2}{MR} \right) + \frac{1}{T} \frac{\beta^3}{2M} \quad (55)$$

944 If we stop at any specific time step with probability $\frac{1}{T}$, and we observe that the average gradient exist
 945 convergent rate $\frac{1}{\sqrt{T}}$ for T large enough which is:

$$\begin{aligned}
E_t \|\nabla l(w_t)\|^2 &\leq \frac{1}{\sqrt{T}}(l(w_0) - l(w^*)) + \frac{1}{\sqrt{T}}\left(\frac{\beta^2}{2M} + \frac{\beta\epsilon^2}{M} + \frac{\beta\sigma_1^2}{MR}\right) + \frac{1}{T} \frac{\beta^3}{2M} \\
&= \mathcal{O}\left(\frac{1}{\sqrt{T}}(l(w_0) - l(w^*)) + \frac{\beta^2}{2M} + \frac{\beta\epsilon^2}{M} + \frac{\beta\sigma_1^2}{MR}\right)
\end{aligned} \tag{56}$$

Situation 2 We now consider the case where the error is propotional to the magnitude of the gradient.
i.e.,

$$E[\|\xi_2\|] \leq \epsilon \|\nabla l(w)\|, \quad \epsilon > 0 \tag{57}$$

We analyze the term as follows:

$$\begin{aligned}
\beta\eta_t^2 E\left[\left\|\frac{1}{M} \sum_{i=1}^M \nabla_S l(w_t + \epsilon_i) - \nabla l(w_t + \epsilon_i)\right\|^2\right] &\leq \beta\eta_t^2 \frac{1}{M} E[\|\xi_1\|^2 + 2\xi_1\xi_2 + \|\xi_2\|^2] \\
&\leq \beta\eta_t^2 \frac{1}{M} \left(\frac{\sigma_1^2}{R} + \epsilon^2 \|\nabla l(w_t)\|^2\right)
\end{aligned} \tag{58}$$

Integrate the term to previous result.

$$l(w_{t+1}) \leq l(w_t) - \left(\eta_t - \frac{\beta\eta_t^2\epsilon^2}{M}\right) \|\nabla l(w_t)\|^2 + \frac{\eta_t(1 - \eta_t\beta)}{2} \beta^2 \sigma_2^2 d + \frac{\beta\eta_t^2\sigma_1^2}{MR} + \beta^3 \eta_t^2 \sigma_2^2 d \tag{59}$$

Set $\eta_t = \eta$. We rearrange and perform same operation and we get:

$$\begin{aligned}
\sum_{t=0}^{T-1} \left(\eta - \frac{\beta\eta^2\epsilon^2}{M}\right) \|\nabla l(w_t)\|^2 &\leq l(w_0) - l(w^*) + \frac{\beta^2\sigma_2^2 d}{2} \sum_{t=0}^{T-1} \eta(1 + \eta\beta) + \frac{\beta\sigma_1^2}{MR} \sum_{t=0}^{T-1} \eta^2 \\
&= l(w_0) - l(w^*) + \frac{\beta^2\sigma_2^2 d}{2} T\eta(1 + \eta\beta) + \frac{\beta\sigma_1^2}{MR} T\eta^2
\end{aligned} \tag{60}$$

Divide by $T(\eta - \frac{\beta\eta^2\epsilon^2}{M})$ on both sides and choose step size such that $(1 - \frac{\beta^2\eta\epsilon^2}{M}) \geq \frac{1}{2}$

$$\frac{1}{T} \sum_{t=0}^{T-1} \|\nabla l(w_t)\|^2 \leq \frac{2}{T} (l(w_0) - l(w^*)) + \beta^2 \sigma_2^2 d (1 + \eta\beta) + \frac{2\beta\sigma_1^2}{MR} \eta \tag{61}$$

We pick $\sigma_2^2 d = \frac{1}{\sqrt{MRT}}$ and $\eta = \frac{\sqrt{MR}}{\sqrt{T}}$ and we will have

$$\begin{aligned}
\frac{1}{T} \sum_{t=0}^{T-1} \|\nabla l(w_t)\|^2 &\leq \frac{2}{T} (l(w_0) - l(w^*)) + \frac{\beta^2}{\sqrt{MRT}} \left(1 + \beta \frac{\sqrt{MR}}{\sqrt{T}}\right) + \frac{2\beta\sigma_1^2}{\sqrt{MRT}} \\
&= \mathcal{O}\left(\frac{1}{\sqrt{MRT}} (2(l(w_0) - l(w^*)) + \beta^2 + 2\beta\sigma_1^2)\right)
\end{aligned} \tag{62}$$

Similar to the first situation, we will have convergence rate with $\frac{1}{\sqrt{MRT}}$ which is $\frac{1}{\sqrt{M}}$ faster than the naive SGD.

Dataset	Corrupt Ratio	Random	Crest	Glister	Craig	Ours
MNIST	0	0.9921±0.0008	0.9892±0.0005	0.9997±0.0001	(*)0.0972±0.0	0.9914±0.0003
	0.1	0.9854±0.0009	0.8384±0.0812	0.7676±0.0271	0.0961±0.0009	0.9876±0.0009
	0.3	0.9772±0.0008	0.3374±0.3385	0.1815±0.0664	0.5196±0.2346	0.9809±0.0025
	0.5	0.962±0.0019	0.3277±0.2062	0.0725±0.0109	0.4264±0.0131	0.9715±0.0021
EMNIST	0	0.8713±0.0026	0.7955±0.0196	0.8246±0.0049	(*)0.0219±0.0	0.8715±0.0015
	0.1	0.8649±0.0016	0.7788±0.0132	0.6762±0.0094	0.0218±0.0007	0.8679±0.0012
	0.3	0.8557±0.0007	0.7621±0.0044	0.4748±0.0236	0.301±0.2553	0.8396±0.0016
	0.5	0.8409±0.0024	0.7146±0.0051	0.2627±0.0172	0.321±0.0111	0.8100±0.0029
CIFAR-10	0	0.8660±0.0015	0.8724±0.004	0.8207±0.0097	0.7618±0.008	0.8757±0.0029
	0.1	0.8481±0.001	0.8440±0.003	0.6350±0.0261	0.7490±0.0066	0.8544±0.0034
	0.3	0.8063±0.010	0.7843±0.004	0.3953±0.0548	0.7050±0.0112	0.8120±0.0058
	0.5	0.7257±0.015	0.6652±0.013	0.2175±0.0292	0.6320±0.0186	0.7318±0.0143
CIFAR-100	0	0.6659±0.0041	0.6728±0.003	0.6004±0.0052	0.5665±0.0053	0.6986±0.0025
	0.1	0.6268±0.005	0.6391±0.004	0.4538±0.0076	0.4629±0.0102	0.6644±0.0034
	0.3	0.5119±0.014	0.5712±0.006	0.2846±0.0074	0.2968±0.0162	0.6085±0.0044
	0.5	0.3293±0.014	0.4305±0.026	0.1578±0.0218	0.1603±0.0048	0.5014±0.0068

Table 4: Performance comparison of different methods across datasets and corruption ratios. Results are reported as mean \pm standard deviation. Each experiments are average over 5 different random seed. The best performance for each setting is highlighted in bold. The * mark the situation in which has diverging behavior during the optimization. The situation usually occurs in Craig method for high corruption scenario.

960 A.5 Discussion about different posterior

961 **Hessian inverse covariance** is the one containing exact information about loss landscape of the
 962 models at certain training points and it is also the posterior used in deriving the theory for sampling.
 963 However, calculation of Hessian can introduce large memory footprint as it will require us to keep
 964 track of the Hessian during training as listed in works Yang et al. (2023). Not only it can cause
 965 large memory footprint, it is also hard to calculate and require steps of approximation. The selection
 966 strategy involving calculation of Hessian will inevitably be slowed down as it requires at least one
 967 forward, backward propagation and the intermediate calculation. Another issue about Hessian inverse
 968 is that despite it contains the curvetures information, it can easily fail to reflect on the true loss
 969 landscape from sampling viewpoint(see 5). For convex optimization view point, Hessian indeed
 970 captures the global information about the loss landscape, but for non-convex region, it can only
 971 express the local curvature information and sampling according to the local information can easily
 972 lead to sampling of high loss region.

973 **Direct training with different random seeds.** The posterior consist of models trained with different
 974 random seeds is studied in Fort et al. (2020) and shown be simple and strong base line for posterior
 975 for uncertainty measurement. The prediction of models trained with different random seeds provides
 976 strong diversity compared to the sampling methods which explore only the local region. In practice,
 977 it is easy to implement and applied to various to different leaning scenarios. The shortcoming of
 978 the method is that it requires much larger memory than other method as it stores and trains multiple
 979 models at the same time. Additionally, it did not necessarily violate the above observation as the
 980 models involved in the posteriors are trained simultaneously and have low loss guarantee.

981 **Diagonal Gaussian** Diagonal Gaussian posteriors is well studied and shown to offer generalization
 982 guarantee. It provides easy control for the region to explore. Despite the fact that it did not contain
 983 local curvetures information, it can still fit in the observation from our theory as long as we select
 984 proper range for the variance. Compared to the two previously mentioned method, it offers better
 985 speed and memory consumption as we only need to sample independent variables for the construction
 986 of the whole distribution. In addition to the advantages mentioned, there is subtle connection between
 987 this posterior and optimization and we will illustrate in the following.

Speed up	CIFAR10	CIFAR100	TinyImagenet
Crest	0.46122396	1.220357534	1.328621908
Our	1.71376899	1.227372798	1.448220165

Table 5: Our method obtain better speed up compare to the benchmark method and the results also generalized to different datasets and architectures. The speed up is calculated as (Full training time / Method training time). The results are average over 5 different random seeds.

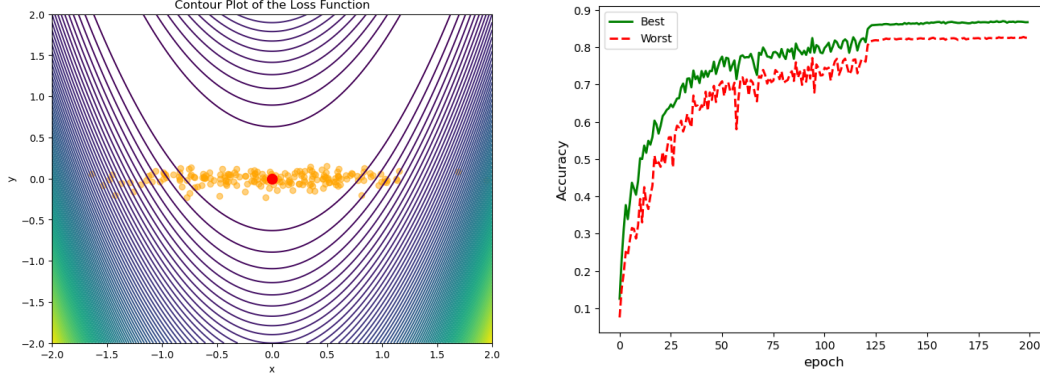


Figure 5: (Left) We calculate the Hessian for sampling at red dot and sampling (yellow dot) using the Hessian inverse as a covariance matrix. The motivation for performing sampling of this kind is that one expect model to be sample lying on the low loss region as it put larger probability density in the small eigenvalue direction. However, when the Hessian loss its ability to represent the true loss curvature (non-convex setting), the Hessian posterior can sample models in regions of high loss, even when the Hessian is computed correctly. (Right) For ensemble method, we find its performance is competitive to Gaussian posterior. However, we find that there exist performance divergence in the different models trained with different random seed and the coresets selected show different performance gain for different models. The coresets selected through this posterior may not be able offer best performance gain in the ensemble.

988 A.6 Details about Greedy selection

989 As pointed out in Mirzasoleiman et al. (2020), one can transform the problem into the following
 990 submodular cover problem with constant C :

$$S^* = \operatorname{argmin}_{S' \subseteq S} |S'| \text{ s.t. } \sum_{j \in S'} \min_{i \in S} \|\nabla l_i(w_t) - \nabla l_j(w_t)\| \leq \epsilon \quad (63)$$

991 The γ in origin problem will be calculated as the number of times a specific sample $j \in S'$ is used
 992 to achieve minimum distance in the argument right hand side in this transformed problem. Greedy
 993 algorithm is used to calculate the sample being selected in which achieve time complexity $\mathcal{O}(nk)$
 994 in existing work Mirzasoleiman et al. (2020); Killamsetty et al. (2021a); Pooladzandi et al. (2022),
 995 where n is the size of the training set and k is the number of samples being selected.

996 Despite linear complexity in terms of the sample selected, the calculation of the difference norm of
 997 the right hand side is still expansive due to the high dimensional properties of deep learning models.
 998 Several works Killamsetty et al. (2021b,a); Mirzasoleiman et al. (2020); Pooladzandi et al. (2022)
 999 demonstrate experimentally and theoretically that one can use the gradient with respect to the last
 1000 layer for the calculation of gradient difference as it captures the norm of difference properly and
 1001 greatly speed up the process for practical application, though a recent work has argued against it.
 1002 Lastly, we complete the formulation with the formulation with fix sample size selected k as following:

$$S^* = \operatorname{argmax}_{S' \subseteq S} C - \sum_{j \in S'} \operatorname{argmin}_{i \in S} \|\nabla l_i(w_t)^L - \nabla l_j(w_t)^L\| \quad (64)$$

s.t. $|S'| \leq k$

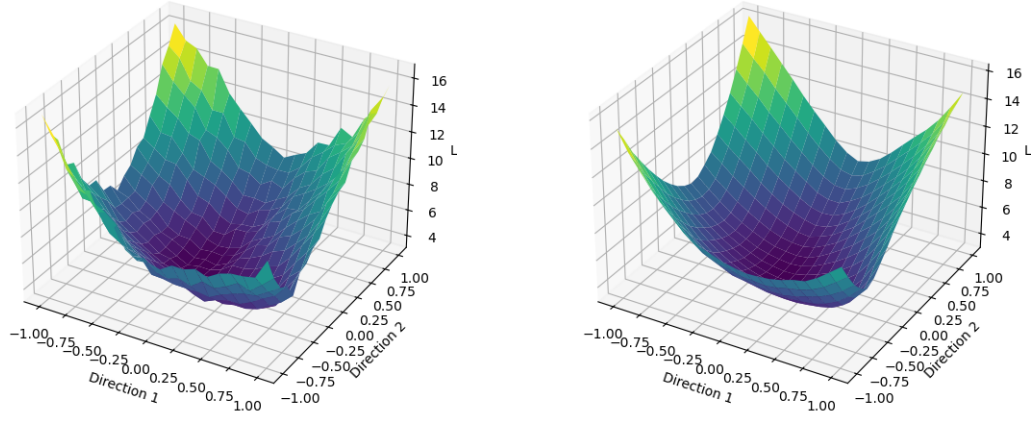


Figure 6: The loss landscape generated by coreset of Craig method and the coreset of our method. The graph is generated with CIFAR10 data and ResNet20 using 1% data budget.

1003 **A.7 Details about the algorithm design**

1004 In this section, we will briefly discuss the design of the our algorithm. First of all, instead of selecting
 1005 the entire training set like Mirzasoleiman et al. (2020), we adapt from Yang et al. (2023) to select
 1006 from mini-batch and union the mini-batch to obtain coreset with specified size. This help to reduce
 1007 the selection time as mentioned in the Yang et al. (2023). For the selection, we calculate the expected
 1008 version as 6 to ensure the properties obtained in the theory remain true. We did not perform threshold
 1009 check listed in the Yang et al. (2023), we instead perform update on each epoch.

1010 A.8 Toy model: Trajectory difference

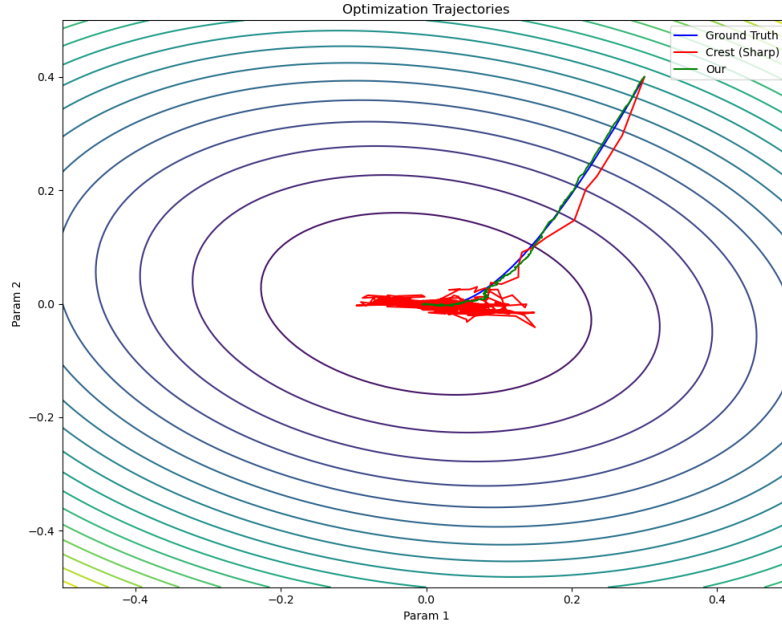


Figure 7: The trajectory difference between gradient descent, Crest and gradient descent. To mimic the gradient mismatch resulting from the label noise, we inject high frequency function into the Crest and our method. To mimic the smoothed version of the loss landscape, we reduce the magnitude of the gradient in our method. We can find that the injected noise can drastically change the trajectory of different methods while the smoothed version can help better recover the ground truth trajectory. The fluctuation from noise becomes more significant around minima as the gradient around minima becomes smaller under this simulated experiment.

1011 A.9 More learning scenarios. (Different training budget, corruption level, and architecture.)

1012 In this section, we perform experiments on even more learning condition such as different corruption
1013 level, training budget, and architecture. For architecture, We extend our result to Vision transformer
1014 to identify whether or not attention based models can work properly with our method and the results
1015 show that our method already outperform others under various structures. For other learning setting,
1016 we also find our method consistently outperform other benchmark which further justify the robustness
1017 of our method.

Budget	Corrupt Ratio	Our Method	Random	Crest
0.2	0.0	0.8969 \pm 0.0026	0.8968 \pm 0.0026	0.8083 \pm 0.0125
	0.1	0.8795 \pm 0.0026	0.8788 \pm 0.0032	0.8331 \pm 0.0454
	0.3	0.8495 \pm 0.0053	0.8483 \pm 0.0027	0.7632 \pm 0.0396
	0.5	0.8022 \pm 0.0020	0.8011 \pm 0.0032	0.5308 \pm 0.0103
0.01	0.0	0.6683 \pm 0.0044	0.6573 \pm 0.0211	0.4370 \pm 0.0184
	0.1	0.6216 \pm 0.0229	0.6147 \pm 0.0272	0.3631 \pm 0.0496
	0.3	0.5667 \pm 0.0037	0.5431 \pm 0.0272	0.3022 \pm 0.0234
	0.5	0.4801 \pm 0.0150	0.4412 \pm 0.0172	0.1988 \pm 0.0231

Table 6: CIFAR-10. Performance comparison of different methods across varying corruption ratios and budget levels. The best-performing method for each setting is highlighted in bold.

Budget	Corrupt Ratio	Our Method	Random	Crest
0.2	0.0	0.7337 \pm 0.0013	0.7291 \pm 0.0010	0.7172 \pm 0.0029
	0.1	0.6712 \pm 0.0022	0.6566 \pm 0.0045	0.6357 \pm 0.0033
	0.3	0.5521 \pm 0.0021	0.5246 \pm 0.0061	0.5081 \pm 0.0064
	0.5	0.4535 \pm 0.0014	0.3884 \pm 0.0059	0.3500 \pm 0.0094
0.01	0.0	0.2660 \pm 0.0079	0.2521 \pm 0.0039	0.1672 \pm 0.0126
	0.1	0.2396 \pm 0.0029	0.2231 \pm 0.0033	0.1469 \pm 0.0039
	0.3	0.1773 \pm 0.0050	0.1566 \pm 0.0025	0.1191 \pm 0.0038
	0.5	0.1235 \pm 0.0026	0.1134 \pm 0.0051	0.0823 \pm 0.0021

Table 7: CIFAR-100. Performance comparison of different methods with varying corruption ratios and budget levels. The best-performing method for each setting is highlighted in **bold**.

Corrupt Ratio	0.0	0.1	0.3	0.5
Our Method	0.8294 \pm 0.0011	0.8077 \pm 0.0008	0.7734 \pm 0.0009	0.7064 \pm 0.0026
Crest	0.8274 \pm 0.0038	0.8002 \pm 0.0042	0.7467 \pm 0.0137	0.6653 \pm 0.0112
Random	0.8200 \pm 0.0034	0.7980 \pm 0.0078	0.7479 \pm 0.0040	0.6806 \pm 0.0030

Table 8: Performance comparison of different selection methods using pretrained ViT-Base on CIFAR-100. Our method consistently outperforms both Crest and random sampling. We train with a learning rate of 0.0003, weight decay of 0.1, and a warm-up scheduling for 20 epochs. The full training process consists of 100 epochs to fit within the rebuttal time constraints.

1018 **B Experiment details**

1019 **B.1 Code base**

1020 We develop our method base on code provided in Crest (Yang et al., 2023). For Craig and Glis-
1021 ter, we use code based from CORD (<https://github.com/decile-team/cords>) with training
1022 hyperparameter changed to our setting.

1023 **B.2 Datasets and architectures**

1024 In our work, we conduct experiments on various image datasets. MNIST (Deng, 2012), EMNIST
1025 (Cohen et al., 2017), CIFAR10, CIFAR100 (Krizhevsky, 2009), and Tinyimagenet (Russakovsky et al.,
1026 2015), SNLI (Bowman et al., 2015) and Imagenet-1k dataset. For MNIST and EMNIST datasets, we
1027 use Lenet. For CIFAR10, CIFAR100, Tinyimagenet and Imagenet-1k, we use respectively ResNet20,
1028 ResNet18 and ResNet50. For SNLI, we use pretrain RoBERTa (Liu et al., 2019) model. To creat data
1029 corruption, we pick specified portion of training samples and flip the corresponding label to other
1030 classes to ensure the corrupt ratio is rigorous. For all experiments except for Tinyimagenet, we run
1031 on single A10 GPU. For Tinyimagenet, we use single NVIDIA A100 GPU.

1032 **B.3 Training hyperparameter**

1033 For all experiments, we fix the peak learning rate at 0.1 and total training epoch to 200. The batch size
1034 is set to 128. For the first 20 epochs, we use linear warm up until learning rate reach 0.1 and decrease
1035 the learning rate by factor 0.1 at 120 epoch and 170 epoch. These hyperparameters were consistent
1036 with those in Yang et al. (2023) and were chosen to ensure fair comparisons across methods and
1037 datasets.

1038 **B.4 Experiment details about loss landscape and its matching**

1039 We generate the loss landscape plot using technics in Li et al. (2018). We purturb the model weights
1040 using

$$f(\alpha, \beta) = l(w^* + \alpha\delta + \beta\eta) \quad (65)$$

1041 In which the δ and η are two randomly initialized vectors with magnitude scaled to models parameters.
1042 The plots are generated using 20 by 20 grid and for each grid we calculate the loss on the whole
1043 training dataset, Craig subset, and our subset with the same parameter. In the plot, we incorporate
1044 our method to the Craig method and use Gaussian noise 0.01. We select all at once as Craig does to
1045 verify that the method will bring smoothness to the loss surface and we observe that there exist more
1046 sharp corner for the loss surface created by Craig and the loss surface generated by our method is
1047 smoother than Craig method. The loss for Craig and our method are scaled loss using the γ constant
1048 obtain through the greedy selection subroutine. Both plot are generated using 1% data budget for
1049 each selection methods. The 3D plot is in Figure 6.

1050 **B.5 Evaluation**

1051 We evaluated different methods on various datasets under different corruption ratios, with a fixed data
1052 budget of 10%. We recorded the final test accuracy and measured time as the process wall time (i.e.,
1053 from the start to the end of the process). To ensure reproducibility, all experiments were conducted
1054 using 5 different random seeds, and results were averaged across these runs.

1055 **C Time complexity analysis**

1056 We appreciate the reviewer’s question and first clarify the notation used in our analysis: The fraction
1057 of data used q . The batch size is m . The number of models to average is M . The number of
1058 parameters in a model is d . The size of the selected subset is R . The total number of data points in
1059 the dataset is n .

1060 We first break down our algorithm into several steps. First, we need to obtain the gradient of samples
1061 in the subset (call the gradient V_p) for M times with noise added to the model weight. This step

will take $\mathcal{O}(MdR)$. Second, we average the gradient and compute the gradient and start the greedy algorithm. The greedy algorithm is of $\mathcal{O}(Rm)$ as we are selecting from R data with m data points selected (which is the same in other methods, such as in Crest). For the details of the greedy algorithm, we refer to (Wei et al., 2014). As we need to select multiple batches, the overall time complexity is $\mathcal{O}(q \frac{n}{m}(Md + Rm)) = \mathcal{O}(q \frac{n}{m}MdR + qn)$.

For CIFAR-10 dataset, for each epoch, the running time for forward pass to obtain the gradient is 0.136 second for each batch, and the running time for the greedy selection is 0.000612 second for each batch. Hence, a much larger time is spent on the forward pass instead of the greedy selection part of the algorithm. The current state of the art method CREST requires expensive tracking of the Hessian which results in significantly longer training time and memory footprint. As a result, our method remains competitive in terms of the total training time while offering improved selection quality.

D Assumptions in the theory

Theorem 4.2 relies on assumptions of third-order smoothness and Hessian symmetry of the loss function

The assumptions of third-order smoothness and Hessian symmetry are commonly used in deep learning theory to facilitate theoretical analysis. One key observation in deep learning is that the loss landscape often exhibits a degree of continuity, meaning that small changes in the parameter space generally lead to gradual changes in the loss function. This aligns with empirical findings on neural network optimization, where sharp transitions in loss are rare under typical training conditions.

The symmetry of the Hessian follows naturally from the continuity and differentiability of the loss function. There are several important results derived using these assumptions. [A, B, C, D, E, F, G](Martens, 2010; Kiros, 2013; Ghorbani et al., 2019; Kunin et al., 2021; Barshan et al., 2020; Yao et al., 2020; Bottou et al., 2018) While non-linear activation functions introduce complexities, prior works suggest that, in practice, the loss function remains smooth enough for such assumptions to be reasonable.[C, H] (Ghorbani et al., 2019; Liu et al., 2023). Additionally, there are lines of research trying to approximate the Hessian using Fisher Information matrix (FIM) such as (Pascanu & Bengio, 2014; Liao et al., 2018; Sen et al., 2024). This implicitly assume that the Hessian is symmetric as FIM is symmetric according to its definition. Also, there are works(Kirkpatrick et al., 2017; Ritter et al., 2018b) using Hessain as a precision matrix in probabilistic models, which implicitly assume symmetry in its structure and receive success in capturing or improving the behavior of deep learning.

Similarly, the third-order smoothness assumption extends this notion by ensuring that second-order derivatives do not change abruptly, which aligns with empirical observations about the optimization dynamics of deep networks. These smoothness and regularity conditions are standard in optimization theory((Jin et al., 2017a; Allen-Zhu & Li, 2018; Carmon et al., 2017; Criscitiello & Boumal, 2021; Jin et al., 2017b; Bottou et al., 2018)) and are widely used to analyze generalization and convergence properties of deep learning models.

Thus, while these assumptions may not hold universally in all settings (and we are not aware of any assumptions that hold universally for all models), they are reasonable approximations that enable theoretical insights into the learning dynamics of deep neural networks. We hope the reviewer agrees that our results are novel and useful within the context of current understanding of deep neural networks.

E Adacore comparison

AdaCore is related to our work in terms of its motivation to capture loss curvature. However, we were unable to include it in our experiments due to the lack of an accessible or functional implementation. We explored multiple sources, including the official AdaCore repository (<https://github.com/opooladz/AdaCore.git>), which has always been empty ever since it was created, as well as public coreset libraries such as CORD (<https://github.com/decile-team/cords>) and DeepCore (<https://github.com/PatrickZH/DeepCore.git>). We did not find a working implementation of AdaCore in any of these repositories.

1113 We also attempted to reimplement the method based on the paper, but were unable to reproduce the
1114 reported performance. The method requires several manually tuned hyperparameters and includes
1115 steps involving Hessian computation, which are both computationally intensive and memory demand-
1116 ing. This introduces a significant runtime and scalability barrier, particularly problematic for the
1117 large-scale or noisy settings we focus on, and undermines the motivation for using coresets to speed
1118 up training.

1119 Additionally, we note that AdaCore has not been included in recent coreset benchmarks, such as
1120 those by (Yang et al., 2023; Okanovic et al., 2023) which includes the authors of Adacore themselves,
1121 where efficiency and scalability are prioritized. We believe this omission reflects a broader consensus
1122 that AdaCore, while conceptually interesting, is not competitive in practice under modern resource
1123 constraints.