Stable Coresets via Posterior Sampling: Aligning Induced and Full Loss Landscapes

Wei-Kai Chang • Purdue University

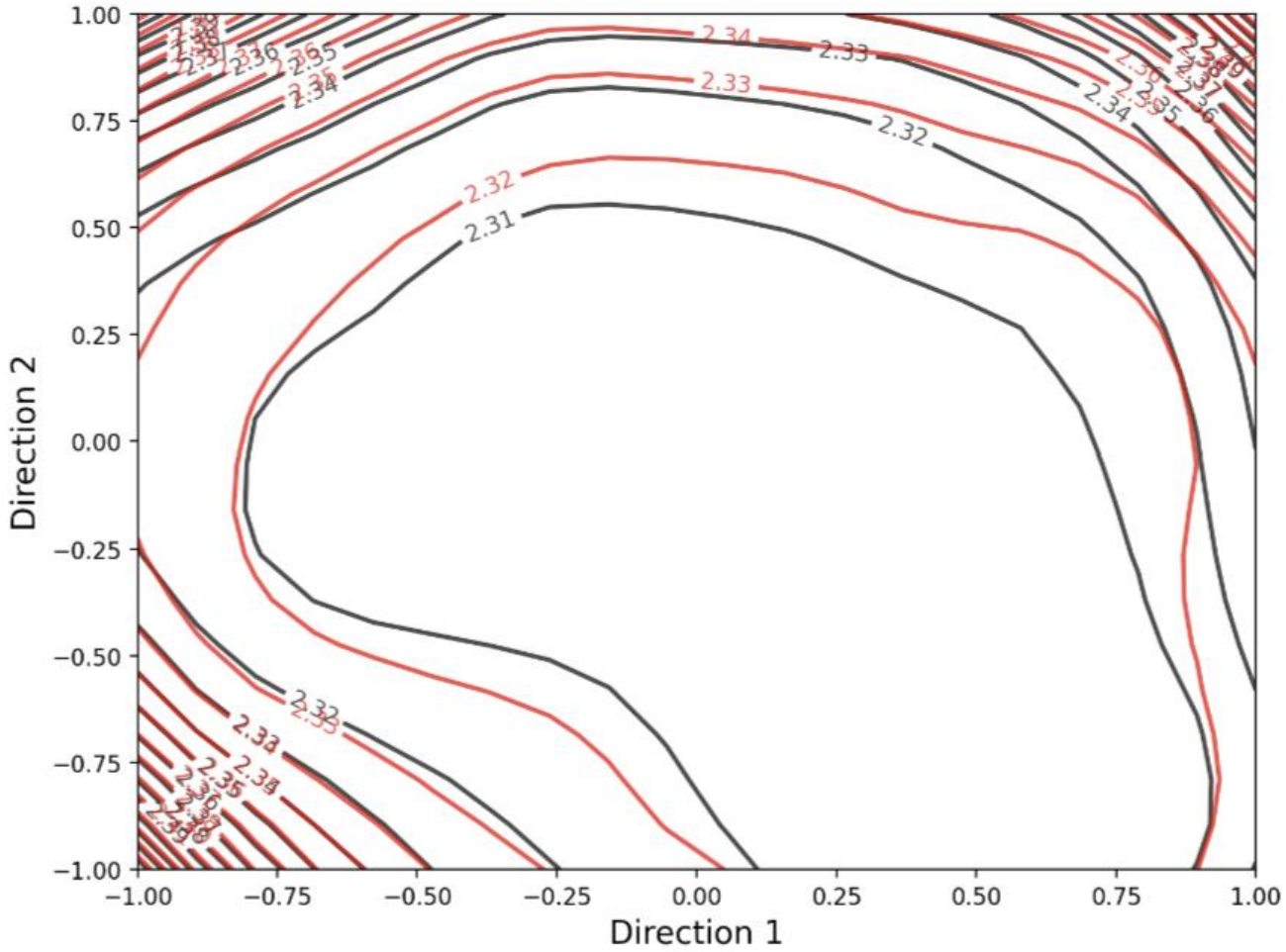
Rajiv Khanna • Purdue University

NeurIPS 2025

Motivation & Problem

- Coreset problems aims to select subset of training set to achieve same training performance as whole dataset.
  - Speed up training
  - Reduce storage of data

- Current methods only match gradient at **current step** and did not imply matching at loss landscape.



Loss landscape by Craig

Key Idea / Intuition

- Instead of selection based on current model weight, we select based on posterior distribution with centering at current weight.
  - Ask the region around current model to also match in terms of gradient!
  - Induced loss landscape match
  - Induced smoothness

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

$$S^* = \underset{S' \subseteq S}{\operatorname{argmin}} |S'| \ \text{ s.t } \sum_{i \in S} \min_{j \in S'} E_\delta ||\nabla l_j(\bar{w} + \delta) - \nabla l_i(\bar{w} + \delta)|| \ \leq \epsilon$$

If a subset matches gradients across different places, it will also better match the loss landscape!!

Method Overview

- We perturb the model weight and average the score across different weights.
- Key components:
  - Simplicity: Gaussian perturbation.
  - Memory: Only at BN layer.
  - Landscape: Sampling multiple times.

Algorithm 1 Ensemble Coreset(r, E, T, B, w_i, η, P)

Parameters: Subsample size R, Ensemble size M, Max epochs T, Number of Batch B, Initial model parameters w_0, Learning rate η, Number of batch selection P, Gaussian Prior δ, minibatch size m
for $t = 1$ to T **do**
  for $p = 1$ to P **do**
    Select random subset $V_p \subseteq S$, $|V_p| = R$
    Select $S_p \in \arg\min_{S_p \subseteq V_p} \sum_{i \in V_p} \min_{j \in S_p} E_\delta ||\nabla l_j(w_t + \delta) - \nabla l_i(w_t + \delta)||$, $|S_p| \leq m$
  end for
  $S_t = \bigcup_{p \in [P]} \{S_p\}$
  for $b = 1$ to B **do**
    Sample batch $S_b \subseteq S_t$, $|S_b| = m$
    $w_{t,b+1} = w_{t,b} - \eta \nabla l_{S_b}(w_{t,b})$
  end for
end for

Theory (Main Result)

- Main theorem: Smoothness and loss landscape match

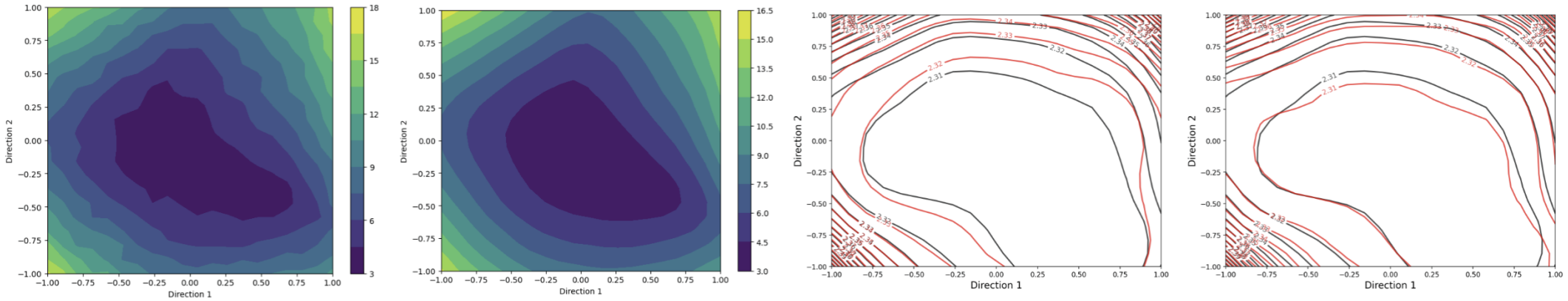
Theorem 3.2. *Suppose a subset $S' \subset S$ is (σ, ϵ, w)-stable and let the Hessian difference be $H_{S',w} - H_{S,w} =: \mathcal{E}$, then,*

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

$$\|\mathcal{E}\| \leq \mathcal{O}(\epsilon^{\frac{1}{2}}) \quad \text{and} \quad \mathrm{tr}(\mathcal{E}^2) \leq \mathcal{O}(\frac{\epsilon}{\sigma}).$$

(2) The difference between newton step of two subset is bounded.

$$\|H_{S',w}^{-1} \nabla l_{S'}(w) - H_{S,w}^{-1} \nabla l_S(w)\| \leq \mathcal{O}(\epsilon^{\frac{1}{2}}).$$



(a) (b) (c) (d)

Theory (Convergence analysis)

- Guarantee convergence with different controlled parameters.

Theorem 3.3. *Say $w \sim N(w_t, \sigma_2 I)$ at epoch t. Assume the $l(\cdot)$ is β-smooth, and the expectation in (6) is calculated by taking M samples. We consider noise in the gradients resulting from the random sampling of batches and coreset selection as ξ_1 and ξ_2 respectively:*

$$\nabla l(w) = \nabla l_S(w) + \xi_1 + \xi_2$$

(1) (Absolute noise) If the noise in coreset selection is of the form:

$$E[||\xi_2||] \leq \epsilon,$$

then by setting the learning rate to be $\eta = \min\{\frac{1}{\sqrt{T}}, \frac{1}{\beta}\}$ and $\sigma_2^2 d = \frac{1}{M\sqrt{T}}$, We can have convergence rate $\frac{1}{\sqrt{T}}$

$$E_t||\nabla l(w_t)||^2 \leq \mathcal{O}(\frac{1}{\sqrt{T}}(l(w_0) - l(w^*) + \frac{\beta^2}{2M} + \frac{\beta\epsilon^2}{M} + \frac{\beta\sigma_1^2}{MR})) \tag{7}$$

2. (Multiplicative noise) If the noise in estimation of gradients is of the form below

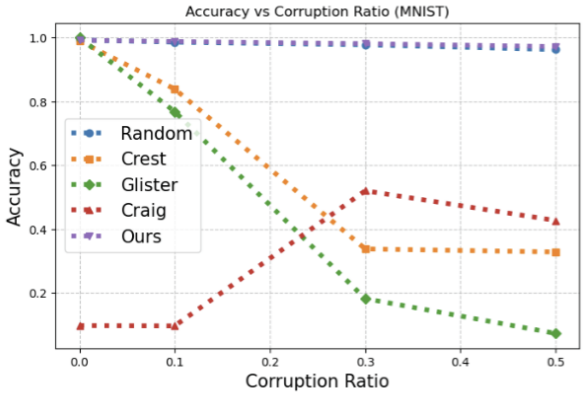
$$E[||\xi_2||] \leq \epsilon||\nabla l(w)|| \tag{8}$$

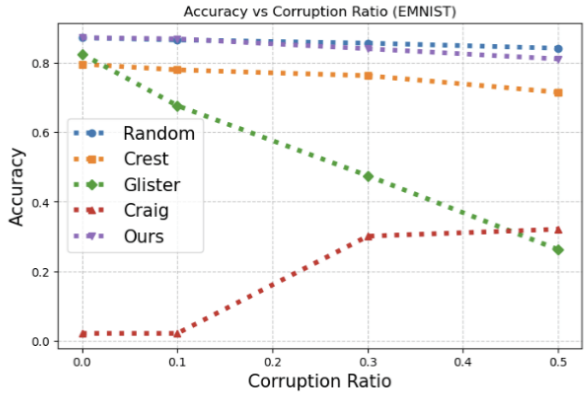
If we set $\sigma_2^2 d = \frac{1}{M\sqrt{T}}$ and $\eta = \min\{\frac{1}{\sqrt{T}}, \frac{1}{\beta}\}$, we can have convergence rate $\frac{1}{\sqrt{MRT}}$

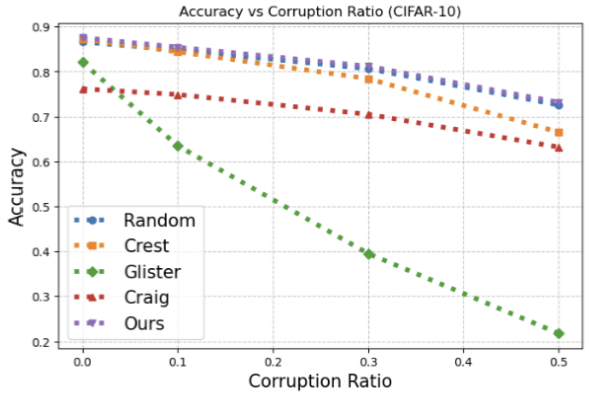
$$\frac{1}{T}\sum_{t=0}^{T-1}||\nabla l(w_t)||^2 = \mathcal{O}(\frac{1}{\sqrt{MRT}}(2(l(w_0) - l(w^*)) + \beta^2 + 2\beta\sigma_1^2)) \tag{9}$$

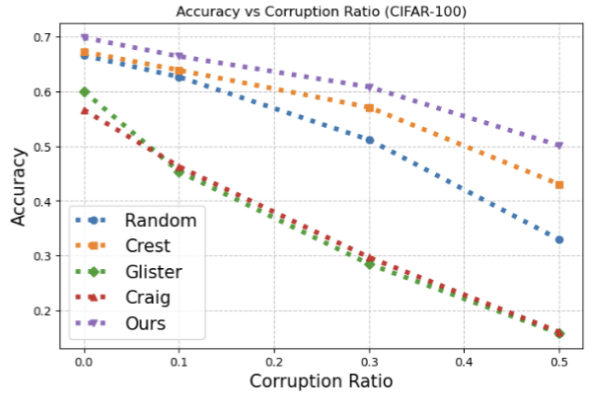
Test performance: Robustness and Accuracy.

Strong across different data corruption

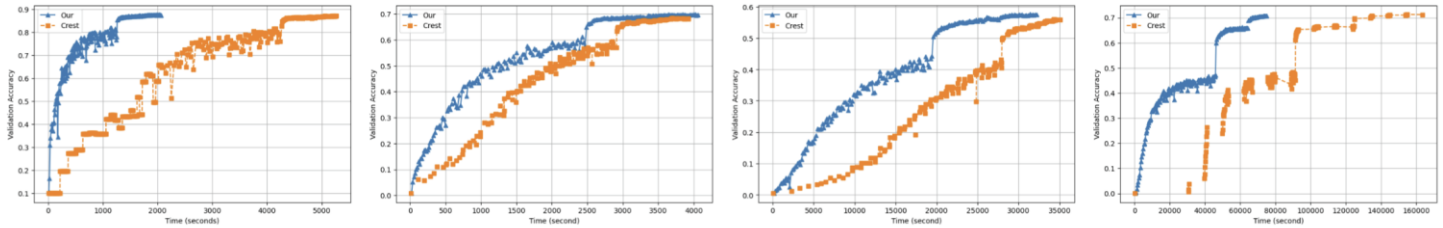








Fast across different datasets



(a) CIFAR-10 (b) CIFAR-100 (c) TinyImagenet (d) Imagenet-1k

Large scale selections

Dataset	Corruption	Our Method	Random	Crest
SNLI	0.0	**0.9132±0.0013**	0.9046±0.0020	0.9098±0.0022
	0.1	**0.8664±0.0012**	0.8324±0.0054	0.8254±0.0028
	0.3	**0.7841±0.0021**	0.7529±0.0031	0.7587±0.0042
	0.5	**0.6062±0.0016**	0.5316±0.0024	0.5104±0.0055
TinyImageNet	0.0	**0.5732±0.0011**	0.5520±0.0094	0.5609±0.0040
	0.1	**0.5526±0.0041**	0.5176±0.0057	0.5150±0.0641
	0.3	**0.4832±0.0043**	0.4193±0.0063	0.4760±0.0061
	0.5	**0.3644±0.0058**	0.2857±0.0137	0.3567±0.0069
ImageNet-1k	0.0	0.7091±0.0004	0.7074±0.0004	**0.7136±0.0015**
	0.1	**0.6977±0.0016**	0.6905±0.0016	0.6946±0.0035
	0.3	**0.6837±0.0007**	0.6514±0.0001	0.6606±0.0024
	0.5	**0.6388±0.0008**	0.5939±0.0017	0.6051±0.0009

Ablation & Insights

Dataset	Corrupt Ratio	Hessian-Gaussian	Ensemble	Spherical-Gaussian
CIFAR-10	0.0	0.8721 ± 0.0035	0.8738 ± 0.0010	**0.8757 ± 0.0029**
	0.1	0.8523 ± 0.0022	0.8506 ± 0.0010	**0.8544 ± 0.0034**
	0.3	0.8041 ± 0.0240	0.8036 ± 0.0010	**0.8120 ± 0.0058**
	0.5	0.5870 ± 0.0156	0.7216 ± 0.0030	**0.7318 ± 0.0143**
CIFAR-100	0.0	0.6841 ± 0.0045	0.6968 ± 0.0070	**0.6986 ± 0.0025**
	0.1	0.6438 ± 0.0043	0.6618 ± 0.0030	**0.6644 ± 0.0034**
	0.3	0.5731 ± 0.0032	0.6066 ± 0.0020	**0.6085 ± 0.0044**
	0.5	0.4340 ± 0.0051	0.4965 ± 0.0060	**0.5014 ± 0.0068**

Table 3: **Sampling with different posterior:** Comparison of posterior sampling methods under different corruption ratios. Bold values indicate best performance. We can observe that for method required accurate Hessian estimation, the performance drop sharper compared to the others.

The Spherical-Gaussian posterior offers a speed-up.

Different random seeds can be parallelized, the increased memory overhead outweighs the benefits, making it a less favorable trade-off.

Hessian-Gaussian is not stable during training.

Layer	0.0	0.1	0.3	0.5
All	0.8654 ± 0.0094	0.8479 ± 0.0081	0.8079 ± 0.0084	0.7288 ± 0.0140
BN	**0.8757 ± 0.0029**	**0.8544 ± 0.0034**	**0.8120 ± 0.0058**	**0.7318 ± 0.0143**
FC	0.8705 ± 0.0019	0.8525 ± 0.0033	0.8099 ± 0.0043	0.7232 ± 0.0171

BN layers consistently yield the best results across different models and datasets giving better trade-off between computational cost and effectiveness.

Summary

- We introduce posterior sampling into the coreset problem and it gives new insight for designing algorithm for the coreset problem!

- Contributions:
  - Robust coreset selection via posterior sampling
  - Extended convergence theory
  - Exhaustive empirical validation

Thank You for watching!