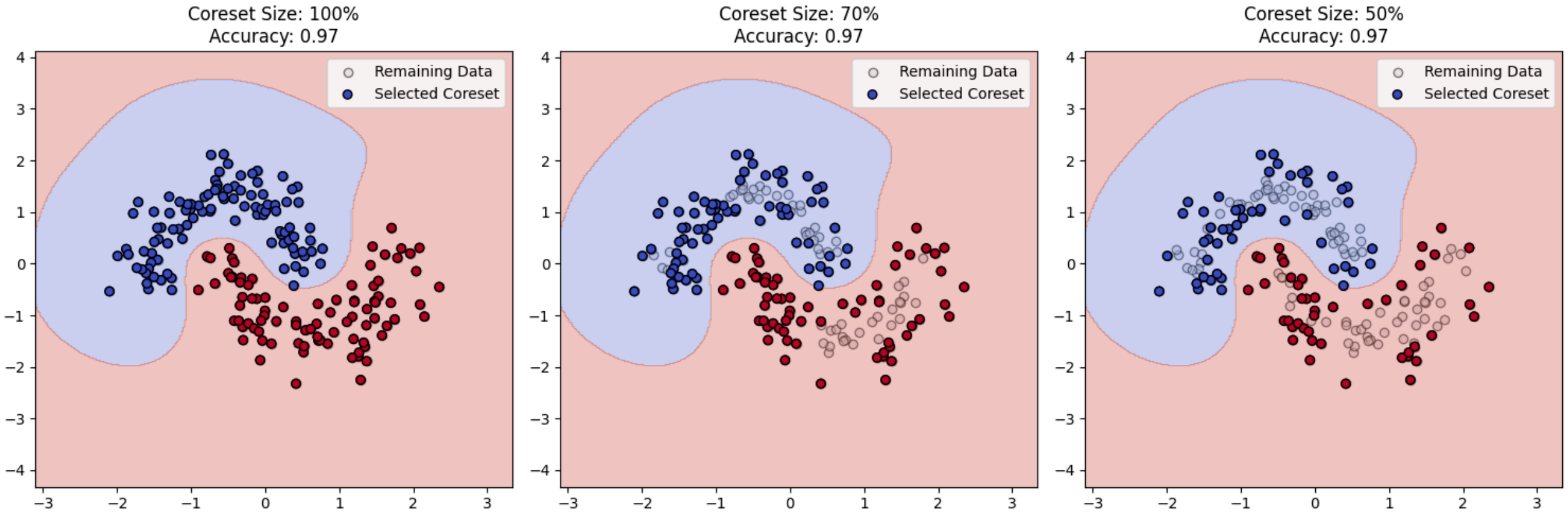
Lightweight Dataset Pruning via Example Difficulty and Prediction Uncertainty

Yeseul Cho*, Baekrok Shin*, Changmin Kang, Chulhee Yun

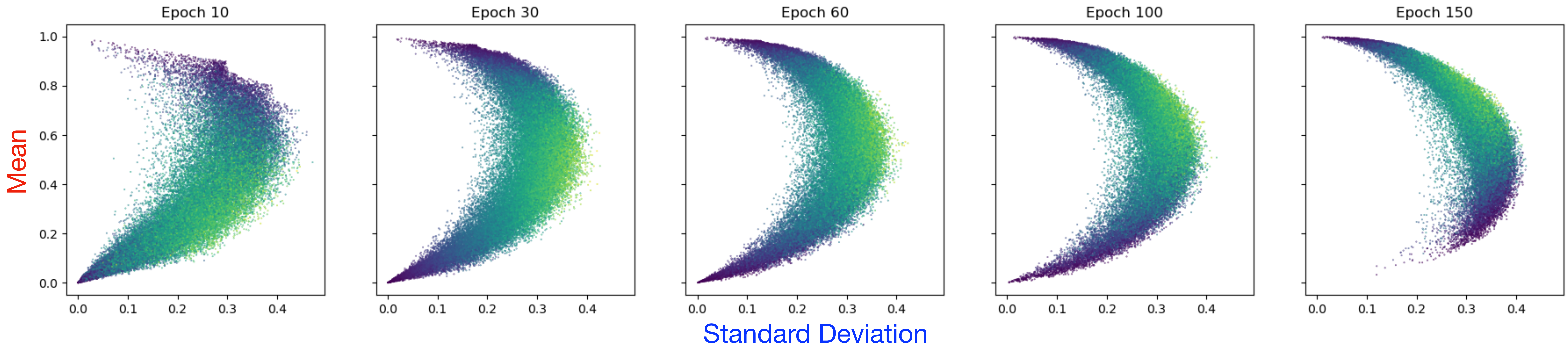
KAIST AI

Dataset Pruning



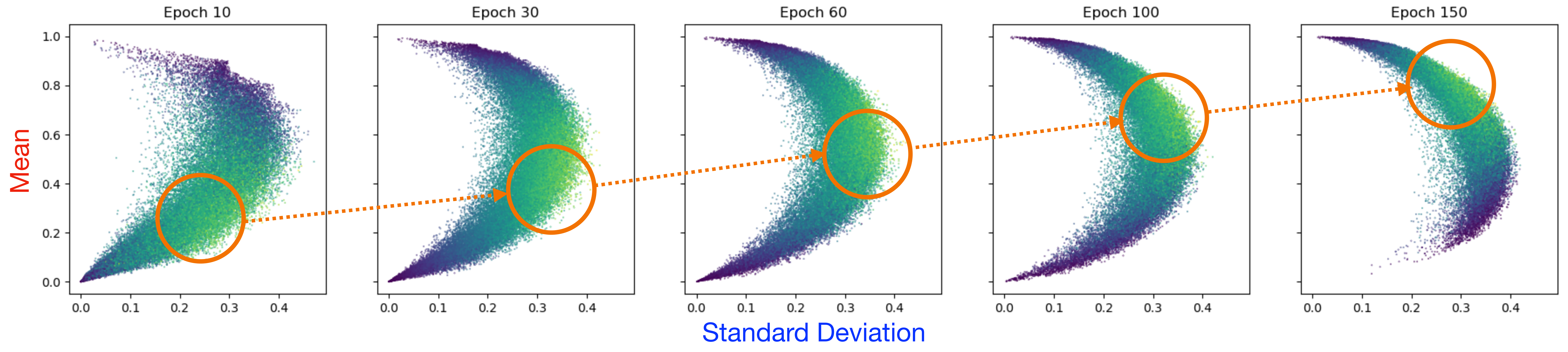
- **Dataset pruning** aims to alleviate storage and training costs by identifying the most informative data points while removing redundant examples.
- However, many existing pruning methods require a **complete training of a model** with a full dataset.
- This ironically makes the pruning process more expensive than just training.

Key Observations



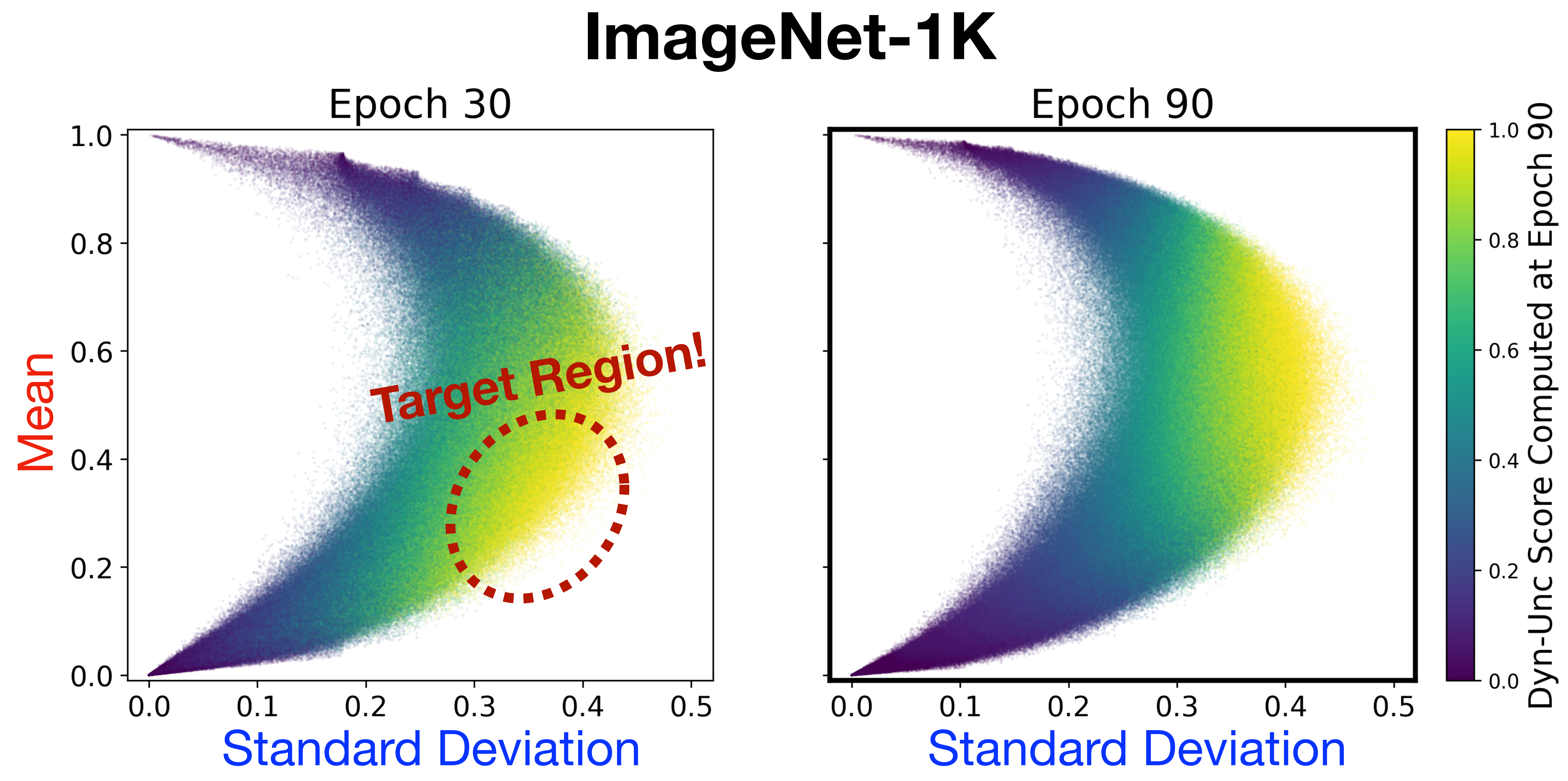
- $\mathbb{P}_k(y \mid \mathbf{x})$: Prediction probability of y given $\mathbf{x}$, for the model trained with k epochs.
- Y-axis: $\bar{\mathbb{P}}(y \mid \mathbf{x}) := \dfrac{\sum_{k=1}^{T} \mathbb{P}_k(y \mid \mathbf{x})}{T}$.
- X-axis: $\sqrt{\dfrac{\sum_{k=1}^{T} \left[\mathbb{P}_k(y \mid \mathbf{x}) - \bar{\mathbb{P}}(y \mid \mathbf{x})\right]^2}{T-1}}$.

Key Observations



The evolution of the data points starts at the bottom left, moves to the right, and ends at the top left as training proceeds

Key Observations



- **`Dyn-Unc`** (He et al., 2024) samples the rightmost part of the “moon plot” by leveraging the standard deviation of the target probability.
- We should target **bottom-right region** to prune the “most uncertain” data points **at earlier epochs.**

Difficulty and Uncertainty-Aware Lightweight Score

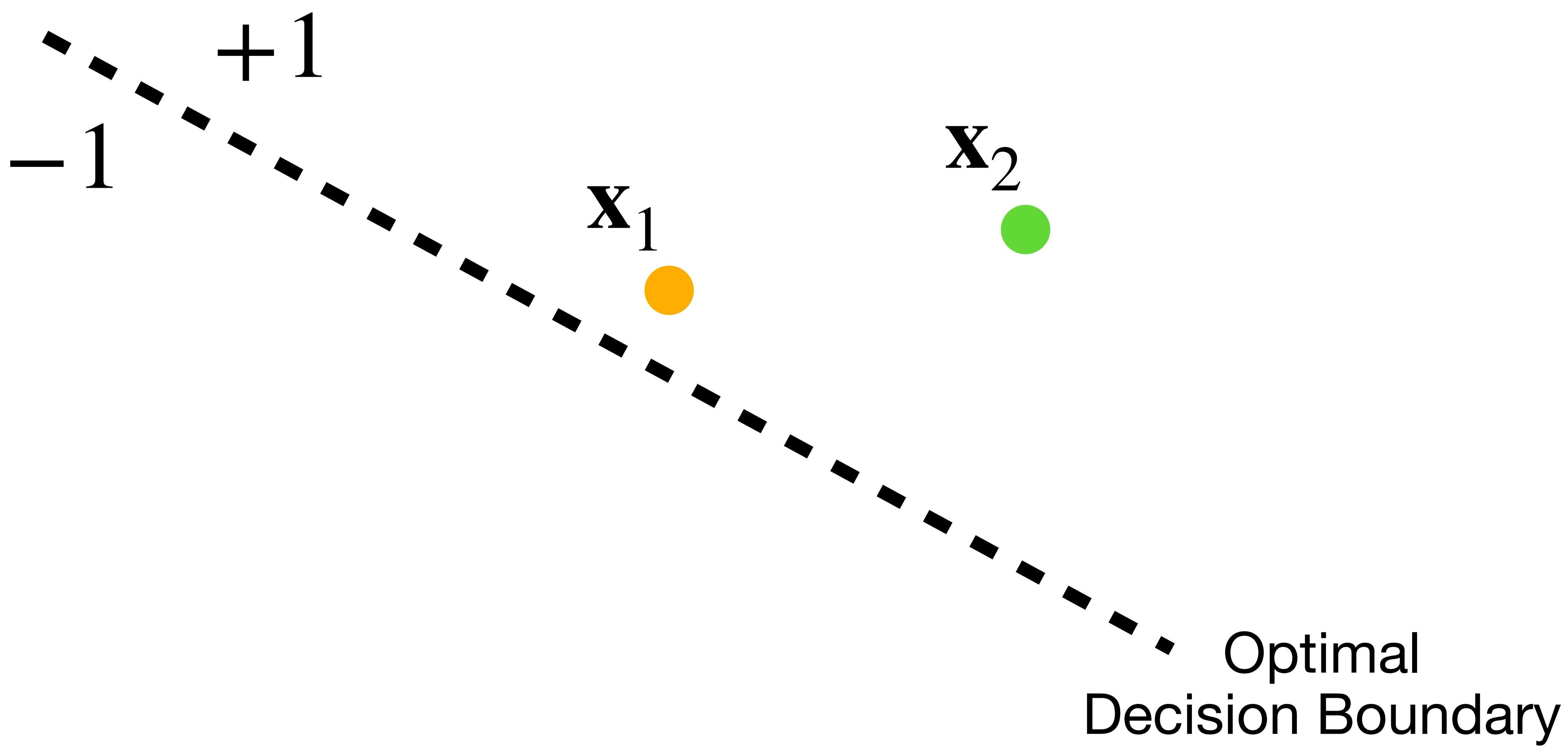
By leveraging example difficulty (y-axis) and prediction uncertainty (x-axis) together,
we can target bottom-right region.

$$\mathrm{DUAL}_k(\mathbf{x}, y) = \underbrace{(1 - \bar{\mathbb{P}}_k)}_{(a)} \underbrace{\sqrt{\frac{\sum_{j=0}^{J-1} [\mathbb{P}_{k+j}(y \mid \mathbf{x}) - \bar{\mathbb{P}}_k]^2}{J-1}}}_{(b)}$$

$$\mathrm{DUAL}(\mathbf{x}, y) = \frac{\sum_{k=1}^{T-J+1} \mathrm{DUAL}_k(\mathbf{x}, y)}{T - J + 1}$$

- Here, $\bar{\mathbb{P}}_k := \frac{\sum_{j=0}^{J-1} \mathbb{P}_{k+j}(y \mid \mathbf{x})}{J}$ is the average prediction over the window $[k, k+J-1]$.
- (a): the example difficulty.
- (b): the standard deviation of the prediction probability, estimating the prediction uncertainty (≈**`Dyn-Unc`**).

Theoretical Analysis



- Consider a linearly separable binary classification task $\{(\mathbf{x}_i \in \mathbb{R}^n, y_i \in \{\pm 1\})\}_{i=1}^{N}$, where $N = 2$ with $\|\mathbf{x}_1\| < \langle \mathbf{x}_1, \mathbf{x}_2 \rangle < \|\mathbf{x}_2\|$.
- Without loss of generality, we assume $y_1 = y_2 = +1$.
- We use a linear classifier, $f(\mathbf{x}; \mathbf{w}) = \mathbf{w}^\top \mathbf{x}$ with a sigmoid activation.
- If only one data point is retained, it should be the one **nearest to the decision boundary**, $\mathbf{x}_1$.

Theoretical Analysis

`Dyn-Unc`: After T_v timestep
`DUAL`: After T_{vm} timestep

where $T_{vm} < T_v$

Theoretical Analysis

Theorem 3.1 (Informal). *Define* $\sigma(z) := (1 + e^{-z})^{-1}$. *Let* $S_{t;J}^{(i)}$ *be the standard deviation and* $\mu_{t;J}^{(i)}$ *be the mean of* $\sigma(f(\mathbf{x}_i; \mathbf{w}_t))$ *within a window from time* t *to* $t + J - 1$. *Denote* T_v *and* T_{vm} *as the first time when* $S_{t;J}^{(1)} > S_{t;J}^{(2)}$ *and* $S_{t;J}^{(1)}(1 - \mu_{t;J}^{(i)}) > S_{t;J}^{(2)}(1 - \mu_{t;J}^{(2)})$ *occurs, respectively. If the learning rate is small enough, then* $T_{vm} < T_v$.

Beta Sampling

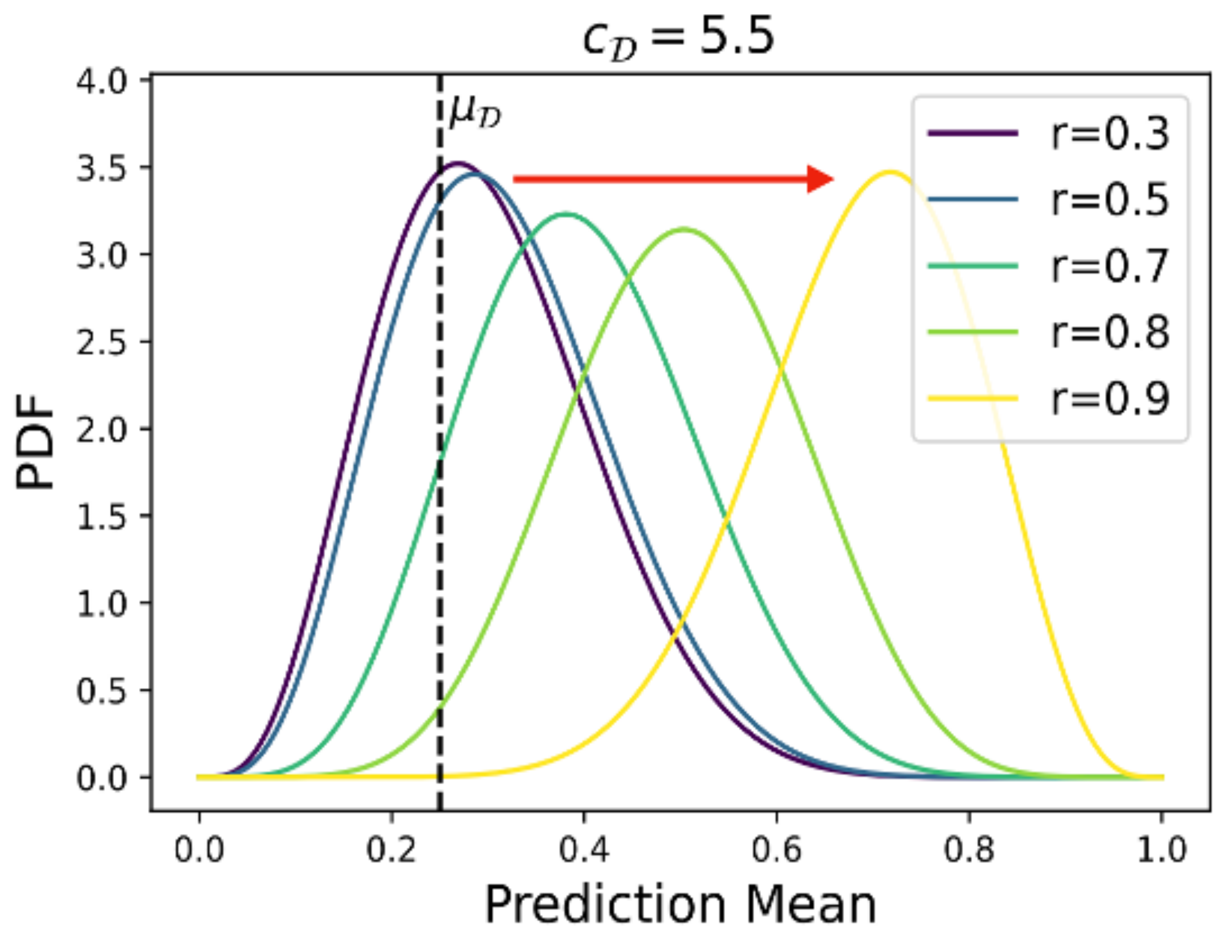
Key Intuition:

The higher the pruning ratio gets,
the more easy samples are needed.

We design the Beta PDF function to assign a sampling probability concerning a prediction mean as follows:

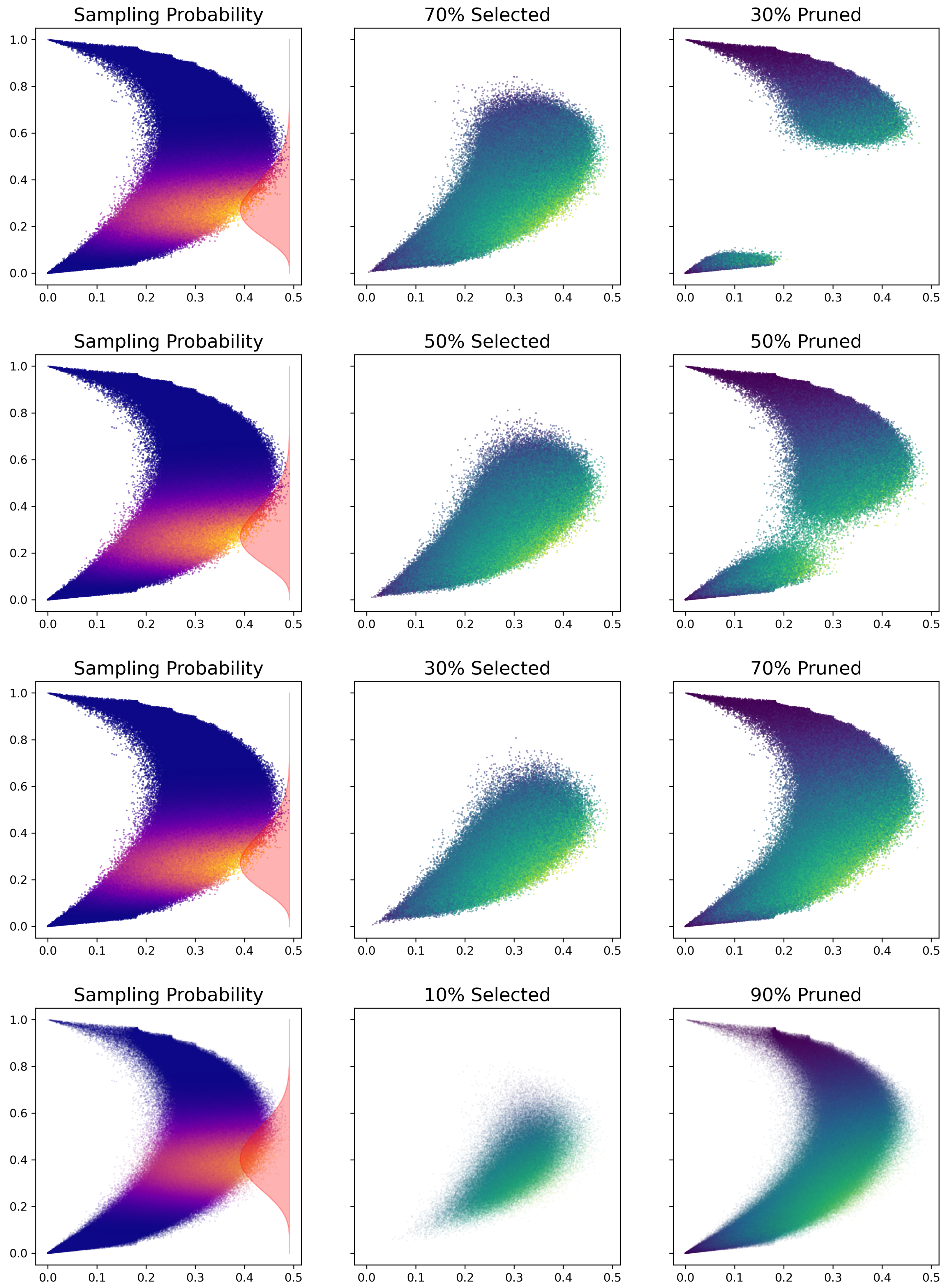
$$\beta_r = 15(1-\mu_D)\cdot(1-r^{c_D}),$$
$$\alpha_r = 15-\beta_r\,.$$

- r : pruning ratio
- μ_D : prediction mean of the highest score sample
- c_D : hyperparameter that determines the nonlinearity

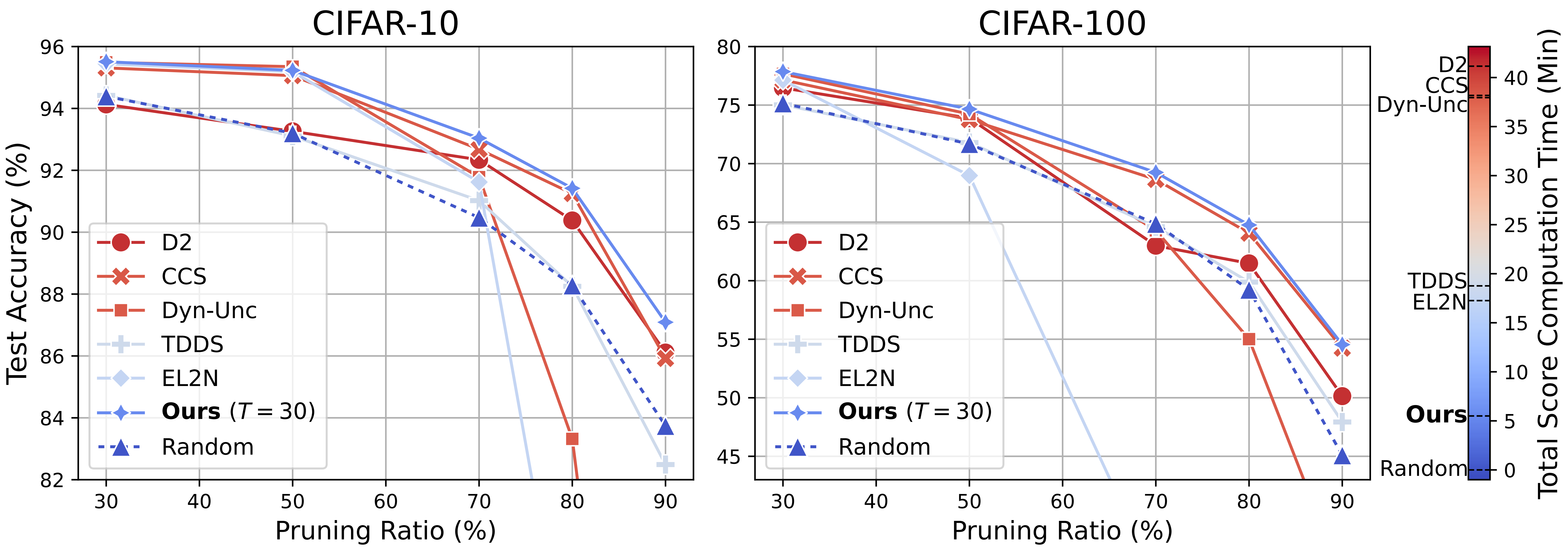


Beta function move progressively with r, starting from μ_D ($r \simeq 0$, small pruning ratio) to one.

Beta Sampling



Experimental Results



ImageNet-1K

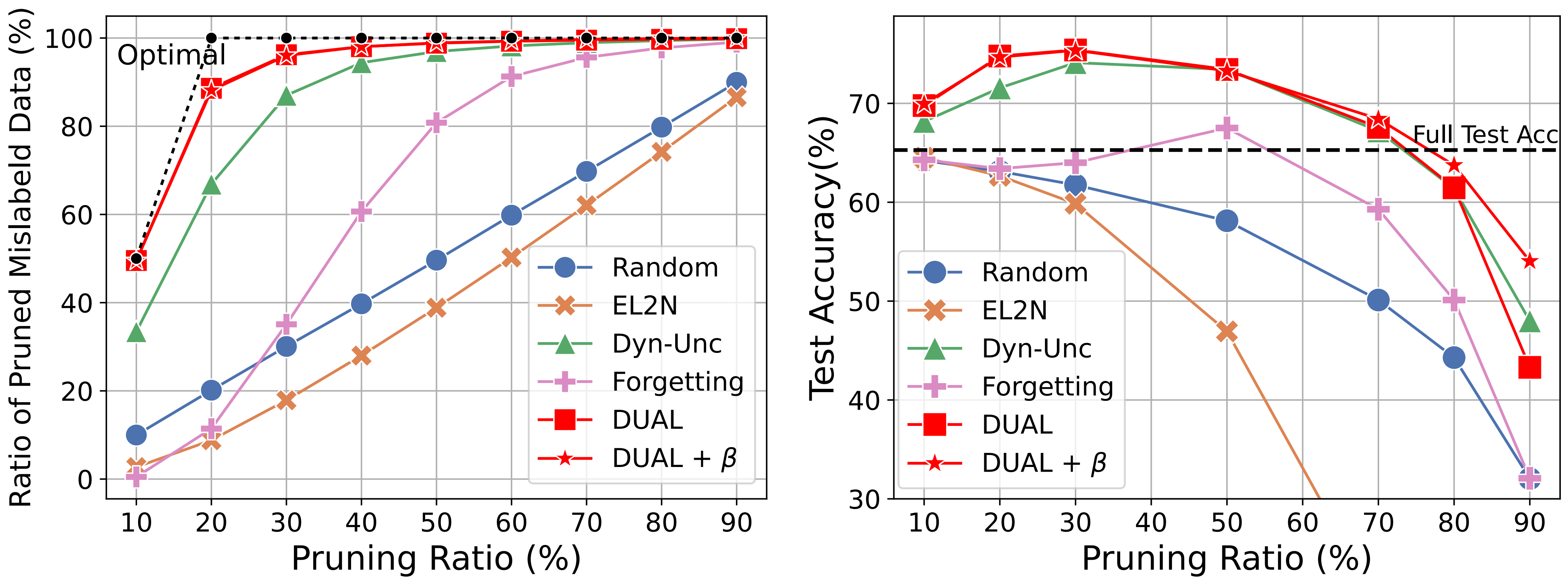
Pruning ratio	30%	50%	70%	80%	90%
`Random`	72.2	70.3	66.7	62.5	52.3
`CCS`	72.3	70.5	67.8	64.5	57.3
`D2`	72.9	71.8	68.1	65.9	55.6
`DUAL`	72.8	71.5	68.6	64.7	53.1
`DUAL+Beta`	**73.3**	**72.3**	**69.4**	**66.5**	**60.0**

$\left.\right\} T = 90$

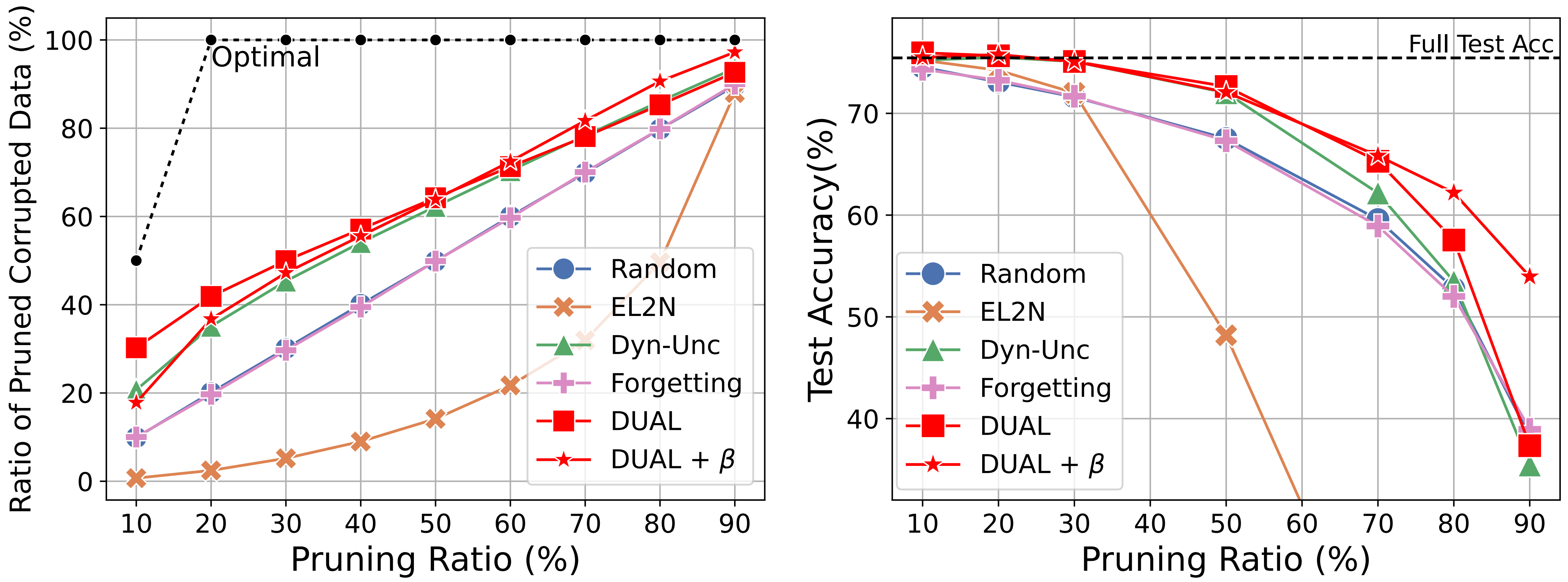
$\left.\right\} T = 60$

Experimental Results

CIFAR-100 Under Label Noise 20%



CIFAR-100 Under Image Corruption 20%



Arxiv

Github