MEASURE, DON'T PRESCRIBE: INFERRING BUDGET-DEPENDENT DIFFICULTY PREFERENCE FOR DATASET PRUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Dataset pruning selects a compact subset to reduce the cost of training on large datasets. Which examples are worth keeping depends on pruning budget: small budgets favor easier, representative examples, whereas larger budgets increasingly favor harder, more informative examples. Existing methods first prescribe a parametric form for the difficulty preference (e.g., hard cutoff, Beta distribution, or contiguous window), and adapt across budgets by adjusting its parameters (e.g., a cutoff ratio, shape parameters, or window position). However, identifying the optimal parameter values at the target budget requires trial-and-error search, and the form is fixed in advance, which may not suit every setting. Instead, we propose **Difficulty Preference Scoring (`DPS`)**, which replaces *prescribe-then-search* with *probe-then-infer*: `DPS` first constructs a difficulty-balanced probing subset, then extracts from its training dynamics the signals that determine how the budget should be adjusted across difficulty levels, and corrects the allocation accordingly. This reduces the adaptation cost at a target budget from extensive trial and error to a single probing run, and lets the preference be inferred from the target setting rather than prescribed in advance. Across a wide range of budgets (1%–50%) and diverse datasets (CIFAR-10, CIFAR-100, QNLI, ImageNet and SQuAD 1.1), `DPS` achieves superior performance over state-of-the-art pruning methods.

1 INTRODUCTION

One-shot dataset pruning aims to construct a fixed, compact subset under a given pruning budget (i.e., the fraction of the original dataset retained), which can be reused across training runs to reduce downstream training and storage costs. A common pruning strategy scores examples by difficulty using training dynamics (Toneva et al., 2019; Pleiss et al., 2020; Coleman et al., 2020; Paul et al., 2021) or model uncertainty (He et al., 2024; Zhang et al., 2024), and retains the hardest ones. However, hardest-first selection degrades severely at small budgets (Sorscher et al., 2022; Zheng et al., 2023; Maharana et al., 2024), revealing a trade-off between informativeness and coverage: hard examples are more informative and offer greater learning value, but are scarce and less representative, whereas easier examples provide broader coverage of the data distribution. Moreover, existing studies reveal that models trained on small subsets are more prone to overfitting hard examples (Dohmatob et al., 2026). Consequently, the difficulty preference depends on the budget: *small budgets favor easier, representative examples, whereas larger budgets increasingly favor harder, informative ones.* Thus, the central challenge is not simply how to score example difficulty, but how the preference across difficulty levels should adapt to pruning budget.

Existing approaches to budget-dependent adaptation fall into several categories: **budget-dependent hard cutoff** discards a fraction of the hardest examples, using a larger cutoff ratio at smaller budgets (Zheng et al., 2023; Maharana et al., 2024; Acharya et al., 2024); **Beta sampling** applies a Beta distribution over difficulty to shift probability mass across difficulty levels (Acharya et al., 2024; Cho et al., 2025); **window-based subset search** slides a budget-sized window along the difficulty ranking and evaluates each candidate (Choi et al., 2024); and other methods introduce **mechanism-specific controls**, such as graph parameters in D^2 (Maharana et al., 2024) and scoring-window parameters in TDDS (Zhang et al., 2024). These methods share a common *prescribe-then-search* paradigm: they define a parametric form $p(x; \theta_r)$ (e.g., a hard cutoff, Beta distribution, or contiguous

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

window) over the difficulty scores x, where its parameters θ_r (e.g., a cutoff ratio, shape parameters, or window position) directly specify which difficulty levels are prioritized at budget r; we refer to these as **preference parameters**. Then, preference is adapted to the target budget by identifying θ_r through trial-and-error experiments (i.e., parameter tuning or subset search).

This paradigm has two limitations. First, it creates a trade-off between adaptation cost and reliable performance: search is expensive, as it requires repeatedly constructing subsets at the target budget and running experiments for comparison. One could choose the preference parameters heuristically, but such choices cannot guarantee reliable performance (Sec. 4.1). Second, a fixed parametric form can limit adaptation: to keep search tractable, adaptation is controlled by only one or a few preference parameters, whose admissible ranges may even be restricted, imposing structural constraints on the resulting allocation (e.g., truncation, a constrained Beta shape, or contiguous concentration). These constraints remain unchanged under tuning and may become restrictive in some settings. We discuss the underlying reasons and resulting failure modes in Sec. 2.

To this end, we propose **Difficulty Preference Scoring (`DPS`)**, which replaces *prescribe-then-search* with *probe-then-infer*: in `DPS`, the difficulty preference is not directly set by a preference parameter, but inferred from a single probing run. `DPS` constructs a difficulty-balanced probing subset at the target budget and measures two complementary signals from its training dynamics: *budget-conditioned difficulty*, which matters more at larger budgets, and *excess fitting*, which matters more at smaller budgets. Since the two signals have opposite budget-dependent properties, their combination infers the difficulty preference and corrects the balanced allocation in a single step.

This paradigm shift offers two advantages. First, adaptation is cheaper: with its configuration fixed across all budgets, `DPS` infers the budget-specific preference from a single probing run rather than extensive trial-and-error experiments, yet consistently outperforms existing methods (Sec. 4.1). Second, adaptation avoids the structural constraints of prescribed forms: since `DPS` infers the preference rather than searching for it, it need not confine the preference to a restricted form that could limit adaptation (Sec. 2). In experiments, we demonstrate the superior performance of `DPS` over state-of-the-art methods across diverse pruning budgets (1%–50%) and benchmarks, covering image classification (CIFAR-10, CIFAR-100, ImageNet), text classification (QNLI), and question answering (SQuAD 1.1). We further evaluate its robustness under different difficulty scores, cross-architecture transfer, and label noise, together with ablation studies.

2 Related Work

Difficulty-based and hybrid dataset pruning. Dataset pruning scores examples by difficulty using training dynamics, such as AUM (Pleiss et al., 2020), EL2N (Paul et al., 2021), and forgetting events (Toneva et al., 2019), or model uncertainty (He et al., 2024; Zhang et al., 2024; Cho et al., 2025). Hard examples are informative but offer limited coverage, motivating hybrid methods that consider both. CCS (Zheng et al., 2023) stratifies examples by difficulty and samples uniformly across strata. D^2 (Maharana et al., 2024) propagates difficulty over a k-NN graph to promote diversity during selection. BOSS (Acharya et al., 2024) combines facility-location-based coverage with Beta-weighted difficulty scores. Although hybrid methods consider both objectives, their relative priority (i.e., the resulting difficulty preference) is controlled by preference parameters.

Prescribe-then-search paradigm for budget adaptation. We unify existing budget-adaptive methods under the same *prescribe-then-search* paradigm: Hard cutoffs exclude a fraction of the hardest examples before selection, with the cutoff ratio adjusted across budgets and datasets (Zheng et al., 2023; Maharana et al., 2024; Acharya et al., 2024). Beta-based methods shift a Beta sampling distribution over difficulty across budgets. Existing approaches impose parameterized constraints on how the Beta shape varies across budgets to control the preference. However, these constraints also restrict the attainable allocations, which could limit adaptation (Acharya et al., 2024; Cho et al., 2025). The window-based method, BWS (Choi et al., 2024), slides a contiguous window along the difficulty ranking, and evaluates each candidate with kernel ridge regression (KRR). However, this KRR proxy is limited to only classification tasks, and other tasks still require model training. Other methods adapt through parameters specific to their own designed mechanisms: D^2 uses a higher message-passing weight and fewer graph neighbors to favor harder examples at larger budgets (Maharana et al., 2024), while TDDS (Zhang et al., 2024) adjusts the length and position of its temporal scoring window to control which difficulty levels are included.

Limited adaptation within prescribed forms. Tuning or search can adapt the preference only within the fixed form, whose structural constraints on the allocation remain unchanged across datasets and budgets. To keep tuning or search tractable, existing methods control the difficulty preference through one or a few preference parameters, whose admissible ranges may be further constrained. This can be restrictive: a hard cutoff cannot adjust the relative priorities among retained difficulty levels; Beta-based methods impose constraints on the Beta shape parameters—BOSS, for example, bounds the mode away from the easy end at every budget; and contiguous windows can group neighboring examples with similar difficulty, reducing subset diversity. Fig. 1 illustrates these structural constraints: CCS produces a truncated uniform allocation, BWS concentrates selection within a short contiguous region, and BOSS cannot place its mode at the easy end even at a 1% budget. These limitations could be relaxed with more preference parameters and broader admissible ranges, but the resulting search cost grows accordingly and soon becomes impractical. Instead, DPS infers the preference from a single probing run, letting measured signals shape the allocation. Further discussion is on Appendix A.6.

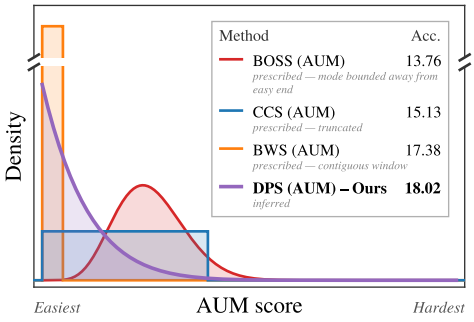


Figure 1: Illustration of the AUM score distribution for each selected subset at 1% budget on CIFAR-100; legend shows test accuracy and allocation structure. All methods use AUM scores.

3 METHOD

Given a dataset $\mathcal{D}$ and a target budget ratio $r \in (0, 1)$, we aim to select a subset $\mathcal{S} \subseteq \mathcal{D}$ of size $r|\mathcal{D}|$ by inferring the difficulty allocation well suited to that budget.

To make this objective tractable, we use difficulty scores (e.g., AUM, EL2N, or Forgetting) obtained from full-data training to partition $\mathcal{D}$ into K strata, $\mathcal{D} = \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_K$, where $\mathcal{D}_k$ denotes the k-th stratum, ordered from easiest ($\mathcal{D}_1$) to hardest ($\mathcal{D}_K$). We then formulate the pruning task as inferring a sampling distribution $p_r = (p_r(1), \ldots, p_r(K))$ over the K difficulty strata, where $p_r(k)$ denotes the fraction of the pruning budget assigned to $\mathcal{D}_k$. The final subset is sampled within each stratum according to this allocation. Thus, p_r directly encodes the difficulty preference at budget r.

3.1 DIFFICULTY-BALANCED PROBING RUN

We perform a single probing run at the target budget, allocating the budget uniformly across strata, $p_{\text{unif}}(k) = 1/K$. The probing subset $\mathcal{P}$ is drawn by filling each stratum's quota $p_{\text{unif}}(k)\, r|\mathcal{D}|$, with any excess quota redistributed across the remaining strata. We then use its training dynamics to infer how the budget should be reallocated across strata, yielding the target distribution p_r.

Why uniform allocation. The probe should measure every stratum without imposing a prior preference; favoring particular strata would bias the probing budget before the preference is inferred. Uniform allocation is also statistically robust: when the noise level of each stratum's signal is unknown, it minimizes the worst-case measurement variance.

Theorem 1 (Informal) — uniform allocation is minimax-optimal. *Suppose stratum k's signal is measured from n_k probed examples with variance ν_k^2/n_k, where the noise levels ν_k are unknown but bounded by a common $\bar{\nu}$. Then, among all allocations satisfying $\sum_k n_k = N$, the uniform allocation $n_k = N/K$ minimizes the worst-case maximum variance $\sup_{\nu_k \le \bar{\nu}} \max_k \nu_k^2/n_k$. (Formal statement and proof: Appendix A.1.)*

Remark (relation to CCS). CCS (Zheng et al., 2023) also uses stratified sampling with uniform allocation, but handles the hard tail differently. It forms strata by equal-width binning of difficulty scores; because these scores are typically long-tailed, most strata lie in the sparse hard tail, causing uniform allocation to over-sample hard examples at small budgets. CCS mitigates this with a hard cutoff, which must be re-tuned for each budget; we instead fold the hardest $\alpha\%$ into the last stratum. We use the same $\alpha\%$ across all budgets and datasets to avoid re-tuning, and Fig. 6 shows that DPS is robust to this choice. The probing subset need not be optimal, but provides a good starting point for one-step correction. As shown in Tab. 8, it performs between the no-cutoff and tuned-cutoff settings, and the final corrected allocation largely improves its performance.

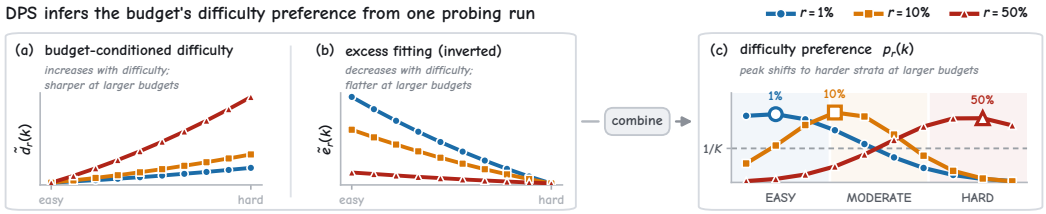


Figure 2: Overview of DPS. The probing run at budget r yields two per-stratum signals with opposite budget-dependent trends: (a) budget-conditioned difficulty and (b) inverted excess fitting. (c) Combining them infers a difficulty preference that shifts from easier to harder strata as the budget increases.

3.2 Difficulty Preference Scoring

From the probing dynamics, DPS extracts two signals that jointly determine the difficulty preference at budget r (Fig. 2): **budget-conditioned difficulty** and **excess fitting**.

3.2.1 Budget-Conditioned Difficulty

Existing difficulty scores are budget-agnostic: they reflect each example's contribution to learning under full-data training, which may differ at the target budget. Thus, we re-measure difficulty from the probing run and define the budget-conditioned difficulty of stratum k as

$$d_r(k) = \frac{1}{|\mathcal{P}_k|T} \sum_{i \in \mathcal{P}_k} \sum_{t=1}^{T} \|f_t(x_i) - y_i\|_2, \tag{1}$$

where $\mathcal{P}_k = \mathcal{P} \cap \mathcal{D}_k$ denotes the probed examples from $\mathcal{D}_k$, f_t is the probe model after epoch t, and T is the total number of training epochs. We observe two key properties of d_r (Fig. 2).

Property (D1). $d_r(k)$ generally increases with stratum k at every budget. This trend is consistent with the easy-to-hard ordering of $\mathcal{D}_1, \ldots, \mathcal{D}_K$: strata with higher original difficulty scores remain harder when re-evaluated at the target budget.

Property (D2). The increase becomes progressively steeper at larger budgets. At large budgets, $d_r(k)$ rises steeply across strata, sharply separating the strata. As the budget decreases, the profile becomes flatter, leaving them weakly separable. We explain this behavior as follows.

With a small subset, fewer examples compete for shared model parameters, so an overparameterized model fits individual examples rapidly, including hard ones (Zhang et al., 2017), narrowing differences across strata. With larger subsets, more easy examples share common patterns, causing their gradients to align and these patterns to be learned earlier (Chatterjee, 2020). As easy examples become fitted, their gradient contributions diminish, while harder examples continue to drive learning, widening the separation in $d_r(k)$. We formalize this intuition as follows:

Theorem 2 (Informal) — larger subsets widen the gap between easy and hard examples. *Consider a linearized model trained with squared loss for a fixed number of epochs on a subset of size N, where each example contains a shared direction and an example-specific direction. The shared direction is fitted at rate $\rho_{\|}(N) \propto N$, as it receives N aligned gradient contributions, while example-specific directions are fitted at an N-independent rate $\rho_{\perp}$. Consequently, trajectory-averaged error decreases for examples aligned with the shared pattern but remains larger for those conflicting with it, and this gap increases with N. (Formal statement and proof, together with proof of Property D2: Appendix A.2.)*

3.2.2 Excess Fitting

At large budgets, d_r can effectively prioritize hard strata; at small budgets, however, its flatter profile provides less separation. We therefore seek a complementary signal from the probing dynamics.

Compared with full-data training, small-subset training is more prone to overfitting on hard examples. Theorem 2 suggests an explanation: easy examples tend to share common, generalizable patterns, whereas hard examples contain rarer, more sample-specific structure. Under full-data training,

aligned gradients from many easy examples reinforce these common patterns, allowing the model to learn them early and build a transferable basis for fitting harder cases. With a small subset, fewer examples reinforce such shared structure, making it easier for the model to fit individual hard examples through sample-specific details and thereby overfit.

This motivates a signal that deprioritizes hard strata at small budgets. For each stratum, we measure how much the probe's fitting fails to translate into generalization, using full-data training as a reference. Specifically, we match the final probe model f_T to the checkpoint along the full-data training trajectory with the most similar generalization behavior; these checkpoints are already available from computing the difficulty scores. We first shortlist the m checkpoints closest in validation accuracy to f_T, then select the one with the closest validation predictions in KL divergence:

$$t_r^\star = \arg\min_{t \in \mathcal{C}} \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{x \in \mathcal{D}_{\text{val}}} \mathrm{KL}(f_T(x) \,\|\, g_t(x)) \,. \tag{2}$$

Since f_T and $g_{t_r^\star}$ generalize comparably, any fitting by f_T beyond the matched checkpoint is excess. We define the excess fitting of stratum k as

$$e_r(k) = \frac{1}{|\mathcal{D}_k|} \sum_{i \in \mathcal{D}_k} \left\| g_{t_r^\star}(x_i) - y_i \right\|_2 - \frac{1}{|\mathcal{P}_k|} \sum_{i \in \mathcal{P}_k} \left\| f_T(x_i) - y_i \right\|_2. \tag{3}$$

Because the probe's final training error is already very small, $e_r(k)$ approximately reflects the error norm of the matched checkpoint on stratum k, i.e., how much of that stratum it has not yet fitted. A larger $e_r(k)$ indicates stronger excess fitting and thus lower priority for that stratum. Since larger $d_r(k)$ instead implies higher priority, we invert $e_r(k)$ for consistency and denote the resulting signal by $\bar{e}_r(k)$. The two signals exhibit **opposite properties** (Fig. 2):

Property (E1). $\bar{e}_r(k)$ generally decreases with stratum k at every budget. This follows the easy-to-hard fitting order of full-data training: easier examples are fitted earlier, whereas harder examples retain larger error norms for longer. Therefore, at any matched checkpoint, the error norm generally increases with k.

Property (E2). The decrease becomes progressively weaker at larger budgets. A small-budget probe matches an earlier stage of full-data training, where easy examples are largely fitted but hard examples are not, causing $\bar{e}_r(k)$ to decrease steeply across strata. When the budget increases, the matched checkpoint shifts to a later stage where most strata are already fitted, making $\bar{e}_r(k)$ nearly flat. A formal proof under Theorem 2 is given in Appendix A.2.

3.2.3 Inferring Difficulty Preference from Signals

We prioritize strata that are both difficult under the target budget and less prone to excess fitting, as captured by high d_r and high $\bar{e}_r$, respectively. We combine the two signals by product so that a stratum receives a high preference score only when both are large.

Concretely, we invert e_r as $\bar{e}_r(k) = \max_j e_r(j) - e_r(k)$ and shift d_r as $d_r(k) - \min_j d_r(j)$, so that both signals are nonnegative and on the same error-norm scale. Then, since adjacent strata have similar difficulty, we smooth the signals to suppress measurement noise (detailed in Appendix A.5). Denoting the smoothed signals by $\tilde{d}_r$ and $\tilde{e}_r$, the preference score of stratum k is their product,

$$s_r(k) = \tilde{d}_r(k) \cdot \tilde{e}_r(k). \tag{4}$$

We standardize s_r across the K strata and map it to a sampling distribution,

$$z_r(k) = \frac{s_r(k) - \mu_r}{\sigma_r}, \qquad p_r(k) = \frac{\exp(\tau\, z_r(k))}{\sum_{j=1}^{K} \exp(\tau\, z_r(j))}, \tag{5}$$

where μ_r and σ_r are the mean and standard deviation of $\{s_r(k)\}_{k=1}^{K}$, and τ is a temperature. Finally, we draw the subset $\mathcal{S}$ from p_r.

The preference is shaped by the measured signals. Since the two signals are nonnegative and exhibit opposing monotonic trends (Properties D1 and E1), their product tends to concentrate preference over a region of the difficulty ranking. The location of this region is determined by the relative

steepness of the two signals (Properties D2 and E2). At small budgets, $\tilde{d}_r$ is nearly flat while $\tilde{e}_r$ decreases steeply, so the product largely follows $\tilde{e}_r$ and favors easier strata. At large budgets, the roles reverse, shifting the preference toward harder strata. Thus, the preferred difficulty moves from easier to harder strata as the budget increases. Appendix A.3 provides a theoretical rationale for the product rule and explains how the two signals jointly determine the allocation, while Tab. 6 provides empirical support with ablation.

Remark (comparison with Beta). The inferred allocation may resemble a Beta distribution with $a, b > 1$, but its shape arises from the measured signals rather than a prescribed Beta form. Existing Beta-based methods adopt a Beta distribution but rigidly constrain its shape parameters in advance: BOSS restricts them to a range where the mode stays away from the easy end; at 1% on CIFAR-100 it favors moderate strata, whereas DPS favors the easiest strata and outperforms it by a margin (Fig. 1). DUAL fixes the sum of the shape parameters to a constant validated only on image datasets, and on the text dataset QNLI, DUAL degrades in performance (Tab. 1). In contrast, DPS does not constrain where its allocation concentrates.

3.2.4 Efficient adaptation across multiple budgets

Variant for Efficient Adaptation. DPS still requires one probing run per target budget. For multi-budget settings, we introduce an efficient variant, $\texttt{DPS}_{\text{amortized}}$. We first probe a few anchor budgets to infer their preferences and then fit a dataset-specific budget-to-preference function. Specifically, we approximate the measured $\tilde{d}_r(k)$ and $\tilde{e}_r(k)$ with quadratic curves and model their coefficients as functions of the budget. Once fitted, preferences at unseen budgets can be directly predicted without additional probing. $\texttt{DPS}_{\text{amortized}}$ still remains competitive with existing methods while enabling more efficient adaptation (Tab. 4). Details are provided in Appendix A.5.1.

4 Experiments

Datasets and Baselines. We evaluate DPS on **five datasets**—CIFAR-10, CIFAR-100 (Krizhevsky, 2009), ImageNet (Deng et al., 2009), QNLI (Wang et al., 2019), and SQuAD 1.1 (Rajpurkar et al., 2016)—covering image classification, text classification, and question answering, at pruning budgets of 1%, 5%, 10%, 20%, and 50%. We compare against **twelve baselines**: six non-adaptive methods with selection strategies fixed across budgets (AUM (Pleiss et al., 2020), EL2N (Paul et al., 2021), Forgetting (Toneva et al., 2019), Entropy (Coleman et al., 2020), Moderate (Xia et al., 2023), and Random), and six budget-adaptive methods (TDDS (Zhang et al., 2024), D^2 (Maharana et al., 2024), CCS (Zheng et al., 2023), BOSS (Acharya et al., 2024), DUAL (Cho et al., 2025), and BWS (Choi et al., 2024)). We use the reported preference-parameter settings when available and otherwise tune them by grid search (Appendix A.7).

Evaluation. For image tasks, we train ResNet-18 (He et al., 2016) on CIFAR-10 and CIFAR-100 for 200 epochs with a batch size of 128, and ResNet-34 on ImageNet for 90 epochs with a batch size of 256. For language tasks, we fine-tune $\text{BERT}_{\text{base}}$ (Devlin et al., 2019) on QNLI for 10 epochs and $\text{GPT-2}_{\text{small}}$ (Radford et al., 2019) on SQuAD 1.1 for 5 epochs, both with a batch size of 64. We halve the batch size at smaller budgets. For all datasets, we sample 5% of the training set for validation and report final model performance on the test set. Full details are provided in Appendix A.4.

Implementation. In DPS, we use AUM as the default score and further evaluate EL2N and Forgetting in Fig. 3. The method configurations of DPS are different from preference parameters: they only affect how the signals are measured and processed, rather than specifying which difficulty levels take priority at each budget, so we keep them fixed across budgets. The stratum number $K = 50$, folding ratio $\alpha = 10\%$, and smoothing settings are also fixed across datasets. Sensitivity analyses of K, α, and τ in Tab. 5 further show that these configurations are less sensitive than baseline preference parameters. Full details are provided in Appendix A.5.

4.1 Adaptation cost

We define adaptation cost as the training runs required to identify the preferred difficulty at a target budget. This is also a one-time, upfront cost to construct a budget-sized subset that can be reused across subsequent training runs and different settings. For DPS, this requires one probing run, cost-

Dataset	Budget	*Non-Adaptive Methods*						*Budget-Adaptive Methods*						
		AUM	EL2N	Forgetting	Entropy	Moderate	Random	TDDS$^{\circ}$	D$^{2\dagger\diamond}$	CCS†	BOSS$^{\dagger *}$	DUAL*	BWS$^{\sharp}$	DPS
CIFAR-10 (Image, Acc.)	1%	16.65 ±0.43	11.51 ±0.32	27.67 ±1.17	14.30 ±1.18	45.40 ±2.06	43.78 ±1.21	38.55 ±1.14	47.29 ±1.71	46.35 ±1.37	41.34 ±0.46	47.94 ±1.34	50.58 ±1.66	**51.36** ±1.13
	5%	30.10 ±0.72	20.47 ±0.32	37.46 ±1.48	35.21 ±2.32	76.51 ±1.62	74.10 ±1.35	70.07 ±1.69	79.38 ±1.39	80.14 ±1.14	**80.97** ±0.13	80.43 ±0.96	78.66 ±1.17	80.95 ±0.74
	10%	46.02 ±0.97	39.30 ±1.43	55.46 ±0.91	63.11 ±1.67	83.01 ±0.63	81.49 ±1.08	83.15 ±1.12	85.02 ±0.17	84.84 ±0.65	85.75 ±0.13	86.01 ±1.32	84.86 ±0.63	**86.44** ±0.15
	20%	79.02 ±0.90	77.47 ±1.40	80.15 ±1.18	87.84 ±0.73	88.00 ±0.21	88.69 ±0.15	90.69 ±0.36	90.65 ±0.09	91.24 ±0.11	91.21 ±0.28	91.57 ±0.17	90.85 ±0.19	**91.68** ±0.12
	50%	94.79 ±0.20	94.41 ±0.20	94.98 ±0.09	93.99 ±0.11	92.54 ±0.09	93.28 ±0.15	94.85 ±0.25	93.06 ±0.10	95.05 ±0.07	95.08 ±0.11	95.04 ±0.14	94.98 ±0.15	**95.14** ±0.08
	Avg.	53.32	48.63	59.14	58.89	77.09	76.27	75.46	79.08	79.52	78.87	80.20	79.99	**81.11**
CIFAR-100 (Image, Acc.)	1%	3.74 ±0.13	3.73 ±0.13	6.50 ±0.09	3.61 ±0.21	9.76 ±0.15	11.12 ±0.87	10.60 ±0.34	14.95 ±0.41	15.13 ±0.20	14.02 ±0.19	15.96 ±0.68	15.95 ±0.88	**18.02** ±0.12
	5%	7.59 ±0.28	6.68 ±0.08	14.41 ±0.16	10.05 ±0.18	26.03 ±1.26	27.33 ±0.22	29.05 ±0.13	37.32 ±1.44	38.68 ±0.58	37.74 ±1.20	40.14 ±0.68	39.05 ±1.09	**40.68** ±0.32
	10%	13.67 ±0.14	11.47 ±0.63	20.93 ±0.39	16.84 ±0.84	42.16 ±0.49	43.63 ±1.73	45.17 ±1.38	53.40 ±0.76	52.16 ±1.41	**54.37** ±0.33	52.43 ±0.39	53.25 ±0.89	54.08 ±0.24
	20%	31.35 ±0.38	31.57 ±1.06	39.65 ±0.24	42.59 ±1.93	60.94 ±0.35	60.16 ±0.67	61.29 ±0.54	64.42 ±0.15	64.70 ±0.96	65.20 ±0.29	63.88 ±0.34	64.04 ±0.13	**65.96** ±0.35
	50%	69.84 ±0.29	70.27 ±0.12	70.80 ±0.26	71.20 ±0.28	72.20 ±0.30	72.12 ±0.11	73.49 ±0.45	74.41 ±0.22	74.04 ±0.31	74.45 ±0.28	74.61 ±0.18	73.51 ±0.29	**75.22** ±0.27
	Avg.	25.24	24.74	30.46	28.86	42.22	42.87	43.92	48.90	48.94	49.16	49.40	49.16	**50.79**
QNLI (Text, Acc.)	1%	30.35 ±1.07	29.63 ±1.77	59.77 ±4.43	56.11 ±4.16	78.16 ±0.64	78.44 ±0.26	34.84 ±5.23	80.94 ±0.53	80.71 ±0.72	77.44 ±2.16	79.93 ±0.41	79.54 ±0.81	**81.04** ±0.55
	5%	21.97 ±1.38	25.13 ±4.13	65.87 ±2.69	47.64 ±2.91	83.49 ±1.09	83.53 ±1.07	28.90 ±2.33	83.47 ±0.84	83.86 ±1.41	81.38 ±1.98	84.01 ±1.72	84.48 ±1.29	**85.12** ±0.67
	10%	21.37 ±1.77	25.08 ±4.42	68.85 ±0.84	48.20 ±3.21	85.55 ±1.00	85.12 ±1.09	40.68 ±1.26	85.42 ±1.20	85.40 ±1.15	**87.52** ±1.42	85.91 ±1.92	85.69 ±1.18	86.60 ±0.84
	20%	27.88 ±0.86	31.13 ±0.28	84.64 ±1.79	54.25 ±2.83	87.85 ±0.36	87.29 ±0.74	50.98 ±1.98	87.61 ±0.58	87.23 ±0.96	88.60 ±1.77	86.06 ±1.85	87.20 ±0.84	**88.85** ±1.11
	50%	89.33 ±1.39	89.19 ±2.03	89.93 ±0.29	89.18 ±1.27	89.76 ±0.10	90.13 ±0.32	90.15 ±0.16	90.69 ±0.26	90.34 ±0.16	90.52 ±0.06	89.05 ±0.70	89.55 ±0.21	**90.70** ±0.20
	Avg.	38.18	40.03	73.81	59.08	84.96	84.90	49.11	85.63	85.51	85.09	84.99	85.29	**86.46**

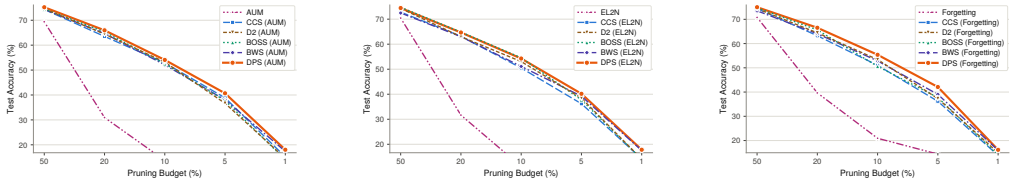
Table 1: Comparison of pruning methods on CIFAR-10/100 with ResNet-18 and QNLI with BERT$_{\text{base}}$. Methods are grouped as **non-adaptive** or **budget-adaptive** according to whether their selection strategy changes with pruning budget. Symbols denote the adaptation mechanism: (†) budget-dependent hard cutoff, (∗) Beta sampling, (♯) window-based subset search, and (⋄) mechanism-specific control. **DPS**† is a variant of DPS adopting the same budget-dependent hard cutoff ratios from baselines. Best result in **bold**, second best underlined.

Dataset	Budget	CCS	BOSS	DUAL	BWS	DPS
ImageNet (Image, Acc.)	1%	12.65	13.92	14.93	11.02	**15.40**
	5%	40.19	42.65	42.12	37.27	**43.26**
	10%	52.39	52.75	54.79	50.51	**55.62**
	20%	61.97	62.04	63.92	63.09	**64.37**
	50%	70.41	71.28	70.91	71.65	**72.05**

Table 2: Comparison against state-of-the-art methods on ImageNet with ResNet-34.

Dataset	Budget	CCS	BOSS	DUAL	BWS	DPS
SQuAD 1.1 (Text, token-level F1)	1%	56.85	55.57	57.04	–	**57.34**
	5%	65.25	62.06	65.49	–	**65.67**
	10%	**67.95**	64.43	67.30	–	67.64
	20%	70.97	67.40	68.97	–	**71.42**
	50%	75.41	74.60	73.60	–	**75.45**

Table 3: Evaluation on QA task (SQuAD 1.1) with GPT-2$_{\text{small}}$. BWS results are missing because its KRR proxy does not apply to QA.



(a) AUM (b) EL2N (c) Forgetting

Figure 3: Comparison under different difficulty scores on CIFAR-100. All methods use the same score, isolating the selection strategy from the score. Baselines tied to their own scoring mechanisms are excluded.

ing 1.15–1.32× a single subset-training run on CIFAR-10 (Tab. 7). For *prescribe-then-search*, this cost is not uniquely defined, as preference parameters may be chosen heuristically or by trial and error. Instead of assigning an arbitrary tuning cost, we compare parameter sensitivity (Tab. 5): performance is highly sensitive to these choices, making tuning necessary for reliable performance. Several strategies can reduce this cost: (i) search with a smaller model and transfer the subset to a larger model, which DPS also supports with better transfer (Fig. 5); (ii) BWS uses cheaper kernel ridge regression proxy to evaluate subsets, but this proxy applies only to classification (Tab. 3); (iii) DUAL amortizes adaptation across budgets through a parameterized budget-to-preference function; similar to DPS$_{\text{amortized}}$; however, DPS$_{\text{amortized}}$ fits this function from a few anchors, each requiring only one probing run and performs better (Tab. 4). Further discussion is in Appendix A.9.

4.2 Benchmark Results

Tab. 1 reports results on CIFAR-10, CIFAR-100, and QNLI, while Tab. 2 and Tab. 3 report ImageNet and SQuAD 1.1 results. For a fair comparison, each baseline adopts the default difficulty score from its original paper. To isolate the effect of the selection strategy, Fig. 3 further compares different methods under the same scores (AUM, EL2N, and Forgetting). From these results, we make several key observations.

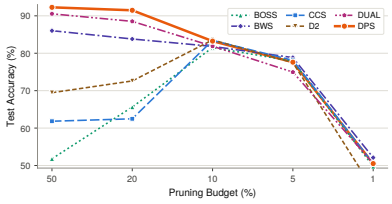
DPS achieves strong performance consistently across budgets, datasets, and difficulty scores. Across all datasets and pruning budgets, DPS achieves the best performance in most settings and ranks second in the remaining. The advantage is particularly clear under aggressive pruning; for example, at a 1% budget on CIFAR-100, DPS improves over the strongest baseline by 2.06 pp. Fig. 3 further shows that the advantage persists under AUM, EL2N, and Forgetting, indicating that it

Dataset	Budget	DUAL	DPS$_{\text{amortized}}$	DPS
CIFAR-10	5%	80.43	80.01	**80.95**
	20%	91.57	**91.78**	91.68
CIFAR-100	5%	40.14	39.48	**40.68**
	20%	63.88	65.46	**65.96**
QNLI	5%	84.01	85.03	**85.12**
	20%	86.06	87.99	**88.85**

Table 4: Efficient multi-budget adaptation at unseen budgets. DPS$_{\text{amortized}}$ uses 1%, 10%, and 50% as anchor budgets to fit a budget-to-preference function, and predicts for 5% and 20% budgets.

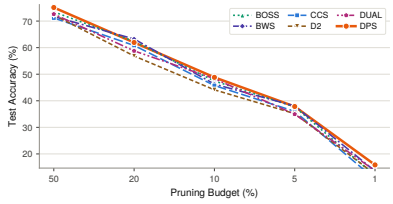
Type	Method / Parameter	RMSD from Default (pp)	CIFAR-100 → ImageNet
Preference parameter	CCS – Hard cutoff	4.44	1.15 pp drop
	DUAL – Beta evolution	2.22	5.09 pp drop
	BWS – Window position	15.39 (2.99)	2.54 pp drop
Method configuration	DPS – Strata K	0.64	Fixed
	DPS – Folding ratio α	0.61	Fixed
	DPS – Temperature τ	1.23	Same on image datasets

Table 5: Sensitivity of preference parameters and DPS's key method configurations on CIFAR-100 at 10% budget. RMSD is measured relative to the default setting: for BWS, 15.39 is computed over all candidate windows and 2.99 over nearby window positions. Transfer reports the performance drop from applying the CIFAR-100 setting to ImageNet.



Label Noise

Figure 4: Comparison against state-of-the-art methods on CIFAR-10 under 20% label noise.



ResNet-18 → ResNet-50

Figure 5: Cross-architecture transfer of subsets selected with ResNet-18 and evaluated with ResNet-50 on CIFAR-100.

is not tied to a particular score. Tab. 1 also includes DPS†, a variant using the same budget-dependent hard-cutoff ratios as baselines. DPS† performs comparably to DPS, suggesting that the improvement mainly comes from the inferred difficulty allocation rather than the choice between hard cutoff and folding. Notably, DPS uses the same configuration across all budgets, which may not be optimal at the target budget (Fig. 6); yet it still performs better than budget-specifically configured baselines.

Existing methods are limited in adapting well across all settings. Non-adaptive methods incur no adaptation cost but perform well only over limited budgets. Budget-adaptive baselines improve over fixed selection rules, yet remain restrictive in some settings. BOSS keeps its Beta mode away from the easy end, limiting its adaptation at aggressive pruning budgets such as 1% and 5%. Under the same hard-cutoff ratios, DPS† variant largely outperforms CCS, D^2, and BOSS because it can reweight the retained difficulty levels. On QNLI, hardest-first methods collapse at most budgets because the hardest are dominated by atypical counter-examples with labels conflict with the common pattern (more in Appendix A.8). TDDS also struggles since its sliding-window variance admits too many such high-variance hard counter-examples. These limitations arise from structural constraints in their selection forms. BWS reduces adaptation cost with a KRR proxy, but this proxy applies only to classification, so BWS is not reported on SQuAD 1.1 (Tab. 3).

4.3 Dataset Pruning in Practical Scenarios

We further evaluate DPS under practical pruning scenarios as follows.

Label noise. Fig. 4 evaluates robustness under 20% label noise on CIFAR-10. We randomly corrupt 20% of the training labels, with all scores and subsets computed on the noisy data. Since mislabeled examples are often mistaken for hard examples, we discard full-data checkpoints after the best validation accuracy when computing excess fitting, as later training increasingly fits noisy data without improving generalization. As shown in Fig. 4, DPS is the most robust across all budgets; among the baselines, DUAL and BWS also show strong robustness.

Cross-architecture transfer. A practical way to reduce the adaptation cost of *prescribe-then-search* is to search with a smaller model and transfer the selected subset to a larger one. The same strategy also applies to DPS. Fig. 5 evaluates transfer from ResNet-18 to ResNet-50 on CIFAR-100 across pruning budgets, where DPS achieves consistently strong performance.

Efficient adaptation to multiple budgets. In multi-budget settings, such as under varying training-resource constraints, adapting separately to every budget can incur substantial overhead. A practical alternative is to infer preferences at a few anchor budgets and fit a budget-to-preference function to predict the preferences at unseen budgets. Following this idea, Tab. 4 evaluates DPS$_{\text{amortized}}$, which uses 1%, 10%, and 50% as anchor budgets and predicts the unseen 5% and 20% budgets. DUAL

Ablation	1%	5%	10%	20%	50%	Avg.
$\tilde{d}_r$ only	26.28 (−25.08)	44.70 (−36.25)	56.37 (−30.07)	80.77 (−10.91)	95.08 (−0.06)	60.64
$\tilde{e}_r$ only	50.34 (−1.02)	76.25 (−4.70)	80.69 (−5.75)	86.84 (−4.84)	93.99 (−1.15)	77.62
normalized sum	43.89 (−7.47)	**81.03** (+0.08)	86.08 (−0.36)	91.26 (−0.42)	92.42 (−2.72)	78.94
DPS ($\tilde{d}_r \cdot \tilde{e}_r$)	**51.36**	80.95	**86.44**	**91.68**	**95.14**	**81.11**

Table 6: Ablation of the scoring design on CIFAR-10. $\tilde{d}_r$ alone performs poorly at small budgets, while $\tilde{e}_r$ alone degrades at larger budgets, consistent with Properties D2 and E2. Their normalized sum fails to recover strong performance at either end.

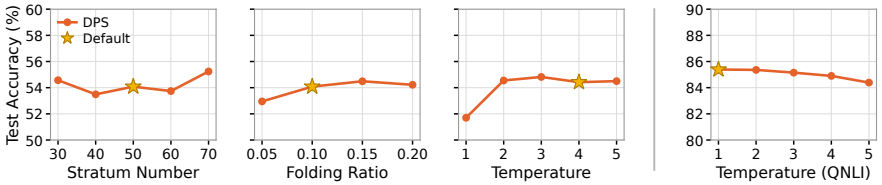


Figure 6: Sensitivity of DPS to its key method configurations on CIFAR-100 at a 10% budget. Stars mark the default values used across all pruning budgets; we also report z-score temperature on QNLI, where a different default is used in text datasets.

similarly parameterizes a nonlinear mapping from the budget to Beta shape parameters. $\text{DPS}_{\text{amortized}}$ outperforms DUAL, while potentially being cheaper since each anchor requires only a single probing run. It performs below DPS, suggesting that a prescribed function form may restrict adaptation.

4.4 Parameter and Ablation Analysis

Tab. 5 compares the sensitivity and cross-dataset transferability of preference parameters with key DPS configurations, while Fig. 6 further examines the sensitivity of DPS across a range of values. Tab. 6 evaluates the scoring design through ablation. We summarize the main observations below.

Pruning performance is highly sensitive to preference parameters, making tuning necessary for reliable performance. Tab. 5 summarizes parameter sensitivity on CIFAR-100 at a 10% budget, where RMSD indicates the performance variation from the default setting across tested values. The preference parameters of CCS, DUAL, and BWS show much larger variation than the configurations of DPS (see Fig. 7 for the baselines and Fig. 6 for DPS). For BWS, RMSD is computed over both its full candidate-window range and the nearby window position. Moreover, we transfer the default CIFAR-100 setting to ImageNet at the same budget and report the performance drop. The preference parameters show weak transferability. DUAL uses a Beta-evolution parameter of 4 on CIFAR-100 and 11 on ImageNet, and the transfer leads to a 5.09 pp drop.

DPS is robust to its key method configurations. Fig. 6 reports the sensitivity of DPS to the stratum number K, folding ratio α, and z-score temperature τ on CIFAR-100 at a 10% budget. K, α, and τ are not preference parameters: they control signal measurement and allocation sharpness rather than specifying which difficulty strata are prioritized. Unlike preference parameters, they are fixed across budgets and show low sensitivity. We use $\tau = 4$ for image datasets and $\tau = 1$ for text datasets due to different data redundancy. Sensitivity to τ on QNLI is similarly low. Notably, most default values are not optimal in the tested settings, yet DPS still outperforms the baselines.

Ablation supports the scoring design. Appendix A.3 provides a theoretical explanation for why we use the product rule and how the two signals jointly determine the allocation. Tab. 6 provides further empirical support: $\tilde{e}_r$ alone performs comparably to the product at the smallest budget, and $\tilde{d}_r$ alone nearly matches it at the largest. Outside these regimes, however, each falls clearly behind. Their normalized sum does not require both signals to be large, so it cannot suppress the less informative signal and degrades at both ends.

5 Conclusion

We introduced DPS, a *probe-then-infer* framework for budget-adaptive dataset pruning that infers difficulty preference from a single probing run rather than prescribing it in advance. By combining budget-conditioned difficulty and excess fitting, DPS adapts the allocation to the target budget with low adaptation cost. Across diverse vision and language tasks and a wide range of pruning budgets, DPS consistently achieves strong performance.

AI USE STATEMENT

Generative AI tools were used to assist with writing and language refinement, figure and table preparation, code development and debugging, mathematical derivations and proofs, and feedback on the presentation and improvement of the work. The research questions, core ideas, methodology, experimental design, and interpretation of results were developed and determined by the authors. All AI-assisted outputs were reviewed and verified by the authors before inclusion.

REFERENCES

Abhinab Acharya, Dayou Yu, Qi Yu, and Xumin Liu. Balancing feature similarity and label variability for optimal size-aware one-shot subset selection. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 96–116, 2024.

Satrajit Chatterjee. Coherent gradients: An approach to understanding generalization in gradient descent-based optimization. In *International Conference on Learning Representations*, 2020.

Yeseul Cho, Baekrok Shin, Changmin Kang, and Chulhee Yun. Lightweight dataset pruning without full training via example difficulty and prediction uncertainty. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10602–10643, 2025.

Hoyong Choi, Nohyun Ki, and Hye Won Chung. BWS: Best window selection based on sample scores for data pruning across broad ranges. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8672–8701, 2024.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations*, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

Elvis Dohmatob, Mohammad Pezeshki, and Reyhane Askari-Hemmat. Why less is more (sometimes): A theory of data curation. In *International Conference on Learning Representations*, 2026.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Muyang He, Shuo Yang, Tiejun Huang, and Bo Zhao. Large-scale dataset pruning with dynamic uncertainty. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 7713–7722, 2024.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. RegMix: Data mixture as regression for language model pre-training. In *International Conference on Learning Representations*, 2025.

Songhua Liu and Xinchao Wang. MGDD: A meta generator for fast dataset distillation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 56437–56455, 2023.

Adyasha Maharana, Prateek Yadav, and Mohit Bansal. $\mathbb{D}^2$ pruning: Message passing for balancing diversity & difficulty in data pruning. In *International Conference on Learning Representations*, 2024.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3428–3448, 2019.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20596–20607, 2021.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q. Weinberger. Identifying mislabeled data using the area under the margin ranking. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17044–17056, 2020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, 2016.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. Beyond neural scaling laws: Beating power law scaling via data pruning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 19523–19536, 2022.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.

Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *International Conference on Learning Representations*, 2023.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

Xin Zhang, Jiawei Du, Yunsong Li, Weiying Xie, and Joey Tianyi Zhou. Spanning training progress: Temporal dual-depth scoring (TDDS) for enhanced dataset pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26223–26232, 2024.

Haizhong Zheng, Rui Liu, Fan Lai, and Atul Prakash. Coverage-centric coreset selection for high pruning rates. In *International Conference on Learning Representations*, 2023.

Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9813–9827, 2022.

A Appendix

A.1 Minimax optimality of uniform allocation

We first consider an idealized estimation model for allocating a probing budget across difficulty strata.

Theorem 1 (Formal version of Theorem 1: minimax optimality of uniform allocation)**.** *Let $N \geq K$ be a probing budget divisible by K. For each stratum k, let $\hat{s}_k$ be the sample mean of $n_k \in \mathbb{Z}_{>0}$ i.i.d. signal measurements with per-example variance ν_k^2, so that $\mathrm{Var}(\hat{s}_k) = \nu_k^2/n_k$. Suppose $0 \leq \nu_k \leq \bar{\nu}$ for a common bound $\bar{\nu} > 0$. Among all allocations satisfying $\sum_{k=1}^K n_k = N$, the uniform allocation $n_k = N/K$ uniquely minimizes the worst-case maximum estimation variance:*

$$\min_{\substack{n_1,\ldots,n_K \in \mathbb{Z}_{>0} \\ \sum_{k=1}^K n_k = N}} \sup_{\boldsymbol{\nu} \in [0,\bar{\nu}]^K} \max_{1 \leq k \leq K} \frac{\nu_k^2}{n_k} = \frac{K\bar{\nu}^2}{N}. \tag{6}$$

Proof. For any feasible allocation, the worst case is attained by setting $\nu_k = \bar{\nu}$ for every stratum, since each term ν_k^2/n_k is increasing in ν_k and the ν_k vary independently. Therefore,

$$\sup_{\boldsymbol{\nu} \in [0,\bar{\nu}]^K} \max_{1 \leq k \leq K} \frac{\nu_k^2}{n_k} = \frac{\bar{\nu}^2}{\min_k n_k} \geq \frac{K\bar{\nu}^2}{N},$$

where the inequality follows from $\min_k n_k \leq K^{-1} \sum_k n_k = N/K$. The uniform allocation attains this lower bound. Since $\bar{\nu} > 0$, equality requires $\min_k n_k = N/K$, which, together with the budget constraint, implies $n_k = N/K$ for every k. Thus the minimizer is unique. □

A.2 Budget-dependent difficulty separation

We use a fixed-feature linear model to study how subset size affects the separation between easy and hard examples during training.

Model. A subset of size N contains $(1-\varepsilon)N$ easy examples with label $y_i = +1$ and εN hard examples with label $y_i = -1$, where $\varepsilon \in (0, \frac{1}{2})$ is fixed across subset sizes. We consider values of N for which both counts are integers.

Let u be a shared unit direction, and let $\{v_i\}$ be orthonormal example-specific directions orthogonal to u. For a fixed $\beta \in (0,1)$, define

$$\varphi_i = \sqrt{1-\beta^2}\, u + \beta v_i.$$

All examples have unit feature norm, and their Gram matrix is

$$\mathbf{K}_N = \beta^2 I_N + (1-\beta^2)\mathbf{1}\mathbf{1}^\top.$$

The model predicts $f_\theta(x_i) = \langle \theta, \varphi_i \rangle$ and is initialized at $\theta_0 = 0$.

Training dynamics. We analyze the deterministic gradient-flow approximation to minibatch SGD with per-example loss $\ell_i(\theta) = \frac{1}{2}(f_\theta(x_i) - y_i)^2$. The minibatch loss is averaged over its B examples, and the batch size B and step size $\eta > 0$ are fixed across subset sizes. In epoch time, the residual vector $\boldsymbol{\xi}_N(t) = (f_t(x_i) - y_i)_{i=1}^N$ follows

$$\dot{\boldsymbol{\xi}}_N(t) = -\frac{\eta}{B}\mathbf{K}_N \boldsymbol{\xi}_N(t), \qquad \boldsymbol{\xi}_N(0) = -\mathbf{y}. \tag{7}$$

The scaling η/B corresponds to N/B updates per epoch: under uniform minibatch sampling, the expected one-step residual increment is $-(\eta/N)\mathbf{K}_N\boldsymbol{\xi}_N$. All subsequent results concern the deterministic flow in equation 7.

Write $\bar{y} = 1 - 2\varepsilon$ and define the two learning rates

$$\rho_\parallel(N) = \frac{\eta}{B}\big[\beta^2 + (1-\beta^2)N\big], \qquad \rho_\perp = \frac{\eta}{B}\beta^2.$$

For a fixed epoch horizon $T > 0$, define

$$d_N(i) = \frac{1}{T}\int_0^T |\xi_{N,i}(t)|\, dt, \qquad F(z) = \frac{1-e^{-z}}{z}, \quad z > 0.$$

Theorem 2 (Formal version of Theorem 2: budget-dependent difficulty separation)**.** *Under the model and dynamics above, the trajectory-averaged errors are constant within each group and satisfy*

$$\begin{aligned} d_N^{\mathrm{e}} &= \bar{y}\, F\big(\rho_{\|}(N)T\big) + (1-\bar{y})\, F(\rho_{\perp}T), \\ d_N^{\mathrm{h}} &= (1+\bar{y})\, F(\rho_{\perp}T) - \bar{y}\, F\big(\rho_{\|}(N)T\big). \end{aligned} \tag{8}$$

As N increases, d_N^{e} strictly decreases and d_N^{h} strictly increases. Consequently, the difficulty gap

$$G(N) = d_N^{\mathrm{h}} - d_N^{\mathrm{e}} = 2\bar{y}\Big[F(\rho_{\perp}T) - F\big(\rho_{\|}(N)T\big)\Big] \tag{9}$$

is positive and strictly increasing in N.

Proof. The Gram matrix $\mathbf{K}_N$ has eigenvalue $\beta^2 + (1-\beta^2)N$ along $\mathbf{1}$ and eigenvalue β^2 on its orthogonal complement. Decompose the initial residual as

$$-\mathbf{y} = -\bar{y}\mathbf{1} + (\bar{y}\mathbf{1} - \mathbf{y}),$$

where the second term is orthogonal to $\mathbf{1}$. Solving equation 7 on these two eigenspaces gives

$$\boldsymbol{\xi}_N(t) = -\bar{y}\, e^{-\rho_{\|}(N)t}\, \mathbf{1} + e^{-\rho_{\perp}t}\, (\bar{y}\mathbf{1} - \mathbf{y}).$$

Thus the residuals of an easy and a hard example are, respectively,

$$\begin{aligned} \xi_N^{\mathrm{e}}(t) &= -\bar{y}\, e^{-\rho_{\|}(N)t} - (1-\bar{y})\, e^{-\rho_{\perp}t}, \\ \xi_N^{\mathrm{h}}(t) &= -\bar{y}\, e^{-\rho_{\|}(N)t} + (1+\bar{y})\, e^{-\rho_{\perp}t}. \end{aligned}$$

Since $0 < \bar{y} < 1$, the easy-example residual is negative. Moreover, $\rho_{\|}(N) > \rho_{\perp}$ implies $\xi_N^{\mathrm{h}}(t) \geq e^{-\rho_{\perp}t} > 0$. Taking absolute values and integrating each exponential using

$$\frac{1}{T}\int_0^T e^{-\rho t}\, dt = F(\rho T)$$

therefore yields equation 8. Subtracting the two expressions gives equation 9.

Finally, the representation $F(z) = \int_0^1 e^{-zs}\, ds$ gives

$$F'(z) = -\int_0^1 s\, e^{-zs}\, ds < 0.$$

Because $\rho_{\|}(N)$ strictly increases with N while $\rho_{\perp}$ remains fixed, equation 8 implies that d_N^{e} decreases and d_N^{h} increases strictly. Their gap therefore increases strictly, and its positivity follows from $\rho_{\|}(N) > \rho_{\perp}$ and the strict decrease of F. □

Corollary 1 (Property D2)**.** *Under the model and dynamics above, the separation between hard and easy examples in trajectory-averaged error, $G(N)$, increases strictly with the subset size N.*

Proof. This is the last assertion of Theorem 2: by equation 9, $G(N) = 2\bar{y}\big[F(\rho_{\perp}T) - F(\rho_{\|}(N)T)\big]$ with F strictly decreasing and $\rho_{\|}(N)$ strictly increasing in N. □

The shared mode has an additional learning rate $\rho_{\|}(N) - \rho_{\perp} = \frac{\eta}{B}(1-\beta^2)N$, while the rate of the example-specific modes remains independent of N. This produces greater difficulty separation at larger budgets despite fixed easy/hard proportions and identical feature norms.

Excess fitting. Let the full data have size N_f and the probe size $N = rN_f$, both with the same ε and trained for T epochs under equation 7. Held-out predictions in this model are governed by the shared direction alone, so we identify the matched checkpoint with the epoch at which full-data training has made the same progress along it as the probe after T epochs,

$$t^\star(N) = T\, \frac{\rho_{\|}(N)}{\rho_{\|}(N_f)},$$

which is strictly increasing in N with $t^\star(N_f) = T$. Excess fitting is the residual norm of the full-data model at $t^\star$ minus that of the probe at T.

Proposition 1 (Property E2)**.** *Under the model and dynamics above,*

$$e_r^{\mathrm{h}} - e_r^{\mathrm{e}} = 2\bar{y}\big[e^{-\rho_\perp t^\star(N)} - e^{-\rho_\perp T}\big],$$

which is strictly decreasing in N and vanishes at $N = N_f$.

Proof. By the proof of Theorem 2, the residual norms of a model trained on M examples at epoch t are $\bar{y}\, e^{-\rho_\parallel(M)t} + (1-\bar{y})\, e^{-\rho_\perp t}$ on easy examples and $(1+\bar{y})\, e^{-\rho_\perp t} - \bar{y}\, e^{-\rho_\parallel(M)t}$ on hard examples. Taking $M = N_f$ at $t^\star$ and $M = N$ at T, the choice of $t^\star$ gives $\rho_\parallel(N_f)\, t^\star = \rho_\parallel(N)\, T$, so the shared-direction terms cancel in both strata:

$$e_r^{\mathrm{h}} = (1+\bar{y})\big[e^{-\rho_\perp t^\star} - e^{-\rho_\perp T}\big], \qquad e_r^{\mathrm{e}} = (1-\bar{y})\big[e^{-\rho_\perp t^\star} - e^{-\rho_\perp T}\big].$$

Subtracting yields the stated difference. Since $t \mapsto e^{-\rho_\perp t}$ is strictly decreasing and $t^\star(N)$ strictly increasing in N, the difference strictly decreases in N, and it vanishes at $N = N_f$ where $t^\star = T$. □

A.3 Rationale for the Product Rule

We record two properties of the preference score $s_r(k) = \tilde{d}_r(k)\tilde{e}_r(k)$ that motivate combining the two signals by a product. Throughout, assume $\tilde{d}_r(k), \tilde{e}_r(k) > 0$.

The product compares relative changes. For consecutive strata,

$$\frac{s_r(k+1)}{s_r(k)} = \frac{\tilde{d}_r(k+1)}{\tilde{d}_r(k)} \cdot \frac{\tilde{e}_r(k+1)}{\tilde{e}_r(k)}.$$

Thus,

$$s_r(k+1) > s_r(k) \quad \Longleftrightarrow \quad \frac{\tilde{d}_r(k+1)}{\tilde{d}_r(k)} > \frac{\tilde{e}_r(k)}{\tilde{e}_r(k+1)}.$$

The score therefore favors a harder stratum exactly when the relative gain in difficulty outweighs the relative loss in inverted excess fitting. A stratum for which either signal is small is also suppressed multiplicatively. In contrast, a sum $\tilde{d}_r + \tilde{e}_r$ compares absolute changes, whose balance depends on the scales of the two signals.

Budget dependence. Define

$$G_r(k) = \frac{\tilde{d}_r(k+1)}{\tilde{d}_r(k)}, \qquad L_r(k) = \frac{\tilde{e}_r(k)}{\tilde{e}_r(k+1)}.$$

Then

$$s_r(k+1) > s_r(k) \quad \Longleftrightarrow \quad G_r(k) > L_r(k).$$

Consistent with Properties D2 and E2, suppose that as the budget grows, $G_r(k)$ increases while $L_r(k)$ decreases. Then the set of strata for which moving harder increases the score,

$$\{k : G_r(k) > L_r(k)\},$$

can only expand with the budget.

If, in addition, $G_r(k)/L_r(k)$ is nonincreasing in k, equivalently $\log s_r(k)$ is discretely concave in k, then s_r is unimodal. Since $G_r(k)/L_r(k)$ increases pointwise with the budget, the smallest maximizer of s_r is nondecreasing with the budget. Thus, under these sufficient conditions, the preferred difficulty shifts toward harder strata as the budget grows.

Monotonicity of the two signals alone does not guarantee unimodality. For example,

$$\tilde{d}(x) = e^{x + \epsilon\sin(\kappa x)}, \qquad \tilde{e}(x) = e^{-x + \epsilon\sin(\kappa x)},$$

with $0 < \epsilon\kappa < 1$, are respectively strictly increasing and strictly decreasing, yet their product

$$\tilde{d}(x)\tilde{e}(x) = e^{2\epsilon\sin(\kappa x)}$$

has multiple local maxima. We therefore only claim a tendency toward a concentrated preference in the main text; smoothing reduces local measurement noise and is not assumed to enforce these conditions.

A.4 Evaluation Details

All methods are evaluated under the same evaluation protocol.

Data splits. For all datasets, we randomly reserve 5% of the training set for validation. The pruning budget is defined with respect to the full training set, i.e., $|S| = r|D_{\text{train}}|$, while the validation split is excluded from the pruning pool. For CIFAR-10 and CIFAR-100, we report performance on the official test sets. For ImageNet, QNLI, and SQuAD 1.1, whose test labels are not publicly available, we use the official validation sets for evaluation.

Image classification. For CIFAR-10 and CIFAR-100, we train ResNet-18 for 200 epochs using SGD with learning rate 0.1, momentum 0.9, and weight decay 5×10^{-4}. The batch size is 32 at 1% and 5% budgets, 64 at 10% and 20%, and 128 at 50%. For ImageNet, we train ResNet-34 for 90 epochs using SGD with learning rate 0.1, momentum 0.9, and weight decay 10^{-4}, with batch size 256 across all budgets.

Language tasks. For QNLI, we fine-tune $\text{BERT}_{\text{base}}$ for 10 epochs using AdamW with learning rate 2×10^{-5} and weight decay 0.01. The batch size is 32 at 1% and 5% budgets and 64 at 10%, 20%, and 50%. For SQuAD 1.1, we fine-tune $\text{GPT-2}_{\text{small}}$ for 5 epochs using AdamW with learning rate 5×10^{-5} and weight decay 0.01, with the same batch-size schedule as QNLI.

A.5 Implementation Details

RMS-gated smoothing. We smooth each signal profile across strata in two steps, both intended to reduce measurement noise. First, we smooth each profile with a moving-window average over neighboring strata. At small budgets, each stratum contains only a few probed examples, so its signal estimate is noisy. Since adjacent strata have similar difficulty, their underlying signals should differ little; fluctuations between them therefore mainly reflect which examples happened to be probed rather than a genuine difference in the signal. Second, we shrink the profile toward its mean when its RMS magnitude is small. Because each profile is shifted to be nonnegative with minimum zero, its RMS magnitude is governed by how much the signal varies across strata. When this variation is very small (e.g., when all strata are well learned), the absolute differences across strata are tiny. However, the product rule compares relative rather than absolute changes (Appendix A.3), thereby amplifying these tiny differences even though they carry little practical information. The gate suppresses such spurious variation before the two signals are combined. Since each shifted profile is zero at one end, both steps also keep that end from zeroing out the product when the profile is nearly flat. Formally, the gating in the second step is given by

$$\tilde{x}(k) = \mu + \lambda\big(x(k) - \mu\big), \qquad \lambda = \frac{\|x\|_{\text{rms}}^{\gamma}}{\|x\|_{\text{rms}}^{\gamma} + \delta^{\gamma}}, \tag{10}$$

where $x = (x(1), \ldots, x(K))$ is a window-averaged signal profile, $\mu = \frac{1}{K}\sum_k x(k)$ and $\|x\|_{\text{rms}} = \sqrt{\frac{1}{K}\sum_k x(k)^2}$ are its mean and RMS magnitude, δ is the gate threshold, and γ is the gate sharpness. We use a smoothing window of 10, a gate threshold of 0.05, and a gate sharpness of 5. These settings are fixed across all datasets and pruning budgets.

Method Configurations. Difficulty scores are computed from full-data training using the same architecture and training epochs as evaluation. We use the same method configuration across all pruning budgets; $K = 50$, $\alpha = 10\%$ and smoothing settings (given above) are also shared across all datasets. We set the checkpoint shortlist size to 10 for CIFAR-10/100, 5 for ImageNet, and 2 for QNLI and SQuAD 1.1, based on the number of available epoch checkpoints (200/200/90/10/5). We set the z-score temperature to $\tau = 4$ for image datasets and $\tau = 1$ for text datasets for suppressing higher redundancy of text data.

A.5.1 Efficient multi-budget adaptation

Efficient multi-budget adaptation. For multi-budget settings, we amortize the probing cost by running `DPS` only at a few anchor budgets. At each anchor, we approximate each measured signal profile over the normalized difficulty rank $x \in [0, 1]$ with

$$\phi(x) = \ell(1 - x) + hx + 4\beta x(1 - x),$$

where ℓ, h, and β characterize the easy end, hard end, and curvature, respectively. The coefficients are fitted by least squares, and each coefficient is then modeled as

$$\theta(r) = a_\theta + s_\theta \log r.$$

For an unseen budget, we reconstruct $\tilde{d}_r$ and $\tilde{e}_r$ from the predicted coefficients and apply the standard DPS pipeline to obtain the preference. Thus, only the anchor budgets require probing, while additional budgets incur no further probing cost.

A.6 Trade-off between flexible allocation and adaptation cost

We define *preference parameters* as parameters that directly determine which difficulty levels are favored at a target budget r. These include θ_r in a prescribed form $p(x;\theta_r)$, as well as parameters that determine θ_r across budgets, such as c_d in DUAL. In principle, increasing the number and admissible range of preference parameters makes the prescribed family more expressive: with sufficiently extensive downstream evaluation, search can approach the empirically optimal allocation within the resulting search space. However, searching over these parameters quickly becomes impractical. For example, with $K = 50$ strata, one could directly treat the K stratum sampling probabilities as preference parameters and search over them through trial and error, but this high-dimensional search would incur prohibitive adaptation cost. Existing *prescribe-then-search* methods therefore restrict the number or admissible range of preference parameters to keep adaptation tractable, at the cost of imposing stronger structural constraints on the resulting allocations. DPS takes a different approach: rather than searching over these degrees of freedom, it infers the K-stratum allocation from a single probing run, **avoiding a low-dimensional prescribed form while keeping adaptation cost low.**

Budget-dependent hard cutoff. CCS first proposes budget-dependent hard-cutoff to remove the hardest examples before selection, and D^2 and BOSS adopt the same cutoff ratios in their implementations. Formally, $\theta_r = c_r$, with $p(x;\theta_r) = 0$ for $x > 1 - c_r$.

Beta sampling. Beta-based methods set $\theta_r = (a_r, b_r)$, with $p(x;\theta_r) \propto x^{a_r-1}(1-x)^{b_r-1}$. BOSS restricts the range of the shape parameters, with $a_r = 1 + \bar{D} + c_a|\mathcal{S}|$ and $b_r = 2 + c_b|\mathcal{S}|$, and grid-searches (a_r, b_r) within this constrained range. DUAL instead defines a parameterized nonlinear mapping $(a_r, b_r) = g(r; c_d)$ from the pruning budget to the shape parameters, so its preference is governed by the dataset-specific parameter c_d. In multi-budget settings, this is efficient because once c_d is identified, no additional adaptation is required. This is analogous to $\text{DPS}_{\text{amortized}}$, for which we fit the budget-to-preference mapping from only three anchor budgets (1%, 10%, and 50%), each with one probing run, and apply to unseen budgets, further reducing the amortized adaptation cost.

Window-based subset search. BWS uses the sliding window position along the difficulty ranking as its preference parameter, i.e., $\theta_r = w_r$ with $p(x;\theta_r) \propto \mathbf{1}[w_r \le x < w_r + r]$, and searches over candidate windows for each target budget. Specifically, it first trains a feature extractor for 20 epochs on a random subset, then evaluates candidate windows using a KRR proxy fitted on the extracted features. This still incurs training cost, and the KRR proxy is limited to classification tasks.

A.7 Baseline tuning

Most budget-adaptive baselines evaluate settings only down to a 10% budget; for missing 1% and 5%, we tune their preference parameters on the validation split toward easier examples, which are preferred at small budgets.

D^2 and BOSS adopt the hard cutoff of CCS, so we use the same cutoff ratio for all three baselines: 50%, 70%, and 50% on CIFAR-10, CIFAR-100, and ImageNet, respectively, at a 1% budget, and 40%, 60%, and 40%, respectively, at a 5% budget. These ratios exceed those used by CCS at its highest pruning rate (30%, 50%, and 30% at a 10% budget).

BOSS constrains its Beta parameters as $a = 1 + \bar{D} + c_a|\mathcal{S}|$ and $b = 2 + c_b|\mathcal{S}|$, with $c_a > c_b > 0$ and $c_a|\mathcal{S}| > 1 - \bar{D}$, yielding a lower boundary of 2 for both shape parameters. Its reported 10% setting already reaches the easy-side boundary of this constrained range, $(a, b) = (2, 2)$. Therefore, we cannot further favor easier examples at 1% and 5% within the BOSS parameter range, and thus use $(2, 2)$ at both budgets. Even at this boundary, the Beta mode remains at moderate difficulty.

TDDS selects examples with high temporal variance within a sliding window. Such variance arises not only from quickly learned easy examples but also from counter-examples (e.g., on QNLI) and

from a few fast-learned classes which could cause class imbalance. TDDS thus performs poorly at 1% and 5% and on QNLI. Shifting the window to include more easy examples, as TDDS suggests, further degrades performance by counter-examples and class imbalance.

For text tasks, we use cutoff ratios of 30%, 20%, 20%, 10%, and 10% at the 1%, 5%, 10%, 20%, and 50% budgets, respectively. For DUAL, we use the dataset-specific parameter $c_d = 3.0$ on QNLI and $c_d = 4.0$ on SQuAD 1.1.

A.8 Failure Mode on QNLI

We observe that QNLI contains a common pattern: examples with high question–sentence overlap are often associated with entailment, whereas the hardest examples are enriched with high-overlap non-entailment counter-examples (McCoy et al., 2019). At smaller budgets, hardest-first methods disproportionately select these counter-examples. With limited data, the model can therefore fit this reversed cue. Since most test examples follow the common pattern, its predictions are systematically inverted, so the accuracy is very low. TDDS exhibits a similar problem, as its high-variance scoring window admits too many hard counter-examples. This failure mode highlights why a fixed selection strategy can be restrictive and why adaptive difficulty preference is an important design problem in dataset pruning.

A.9 More on adaptation cost

One-shot (static) and dynamic dataset pruning target different forms of efficiency. Dynamic pruning aims to reduce the cost of model training by deciding during training which examples to use, typically based on training-time signals. In contrast, one-shot pruning aims to obtain a fixed subset that can be reused across training runs and different settings, allowing its construction cost to be amortized over subsequent use. Its efficiency objective therefore typically concerns the cost of constructing a reusable subset, rather than the end-to-end cost of a single model-training run.

In this setting, difficulty scoring is an upfront cost that can likewise be amortized over subsequent use. When difficulty scoring requires training on the full dataset, a single-run end-to-end comparison is less meaningful, since one could simply use the resulting model instead of training another model on the selected subset. We define adaptation cost as an additional upfront cost required to determine the difficulty preference for a budget-sized subset. Similar upfront costs arise in data mixing, where candidate mixtures are evaluated with smaller models before large-scale training (Liu et al., 2025), and in dataset distillation, which incurs substantial computation to construct a reusable synthetic dataset (Zhou et al., 2022; Liu & Wang, 2023). Nevertheless, adaptation cost remains important because it is incurred for every dataset–budget setting.

Under this definition, heuristic choices or fixed selection rules incur no adaptation cost but offer no guarantee of reliable performance, trial-and-error search over k candidates costs k subset-training runs, and `DPS` requires a single probing run.

To quantify the adaptation cost of `DPS` more precisely, Table 7 reports the GPU-hours of one probing run relative to training on a subset at the same budget on CIFAR-10. Across budgets, one probing run costs 1.15–1.32× one subset-training run, i.e., roughly the cost of a single subset-training run.

Budget	**1%**	**5%**	**10%**	**20%**	**50%**
GPU-hour ratio	1.15×	1.25×	1.31×	1.32×	1.32×

Table 7: GPU-hour cost of `DPS` adaptation on CIFAR-10 relative to one subset-training run at the same budget.

A.10 Extra Results

Method	1%	5%	10%	20%	50%
CCS w/o cutoff	22.49	43.77	56.56	86.10	95.05
Probing subset (fixed 10% folding)	35.99	76.16	84.43	91.28	93.74
CCS w/ tuned cutoff	46.35	80.14	84.84	91.24	95.05
DPS	**51.36**	**80.95**	**86.44**	**91.68**	**95.14**

Table 8: Performance of the probing subset on CIFAR-10.

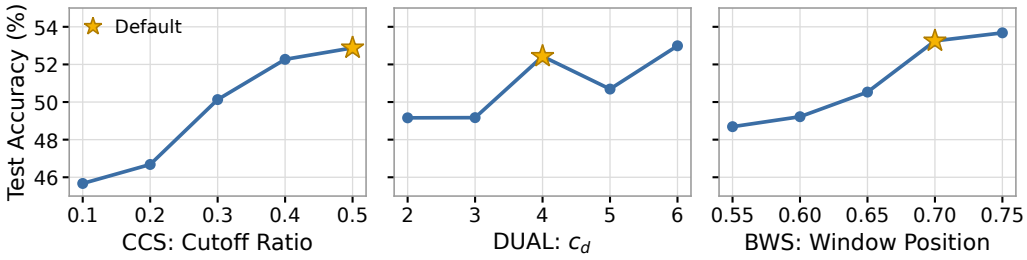


Figure 7: Sensitivity of baselines to their preference parameters on CIFAR-100 at a 10% budget. Stars mark the values used in their papers. For BWS, we report only nearby window positions rather than the full search range.