SCALABLE, TRANSFERABLE META-NETWORK FOR DATA SELECTION REQUIRES A DIFFERENT LOSS (AND WHY THE OBVIOUS CHOICE IS PROBLEMATIC)

Anonymous authors
Paper under double-blind review

ABSTRACT

Data selection is critical for training large language models on massive and heterogeneous corpora. Meta-learning for Training-data Selection offers a principled alternative to heuristic scoring by learning data weights from a target validation objective, but existing methods face a trade-off between fine-grained valuation and transferability to unseen data. A natural solution is to replace per-sample weights with a selection network. However, we find that directly incorporating such a network into existing MTS objectives leads to unstable optimization and poor generalization, caused by weight suppression and persistent reliance on easy-to-learn features. To address these issues, we propose Transferable Example Scoring and Selection (TESS), a scalable data-selection framework built on a Pointwise Value Matching objective (PVM). Experiments on LLM safety and targeted instruction tuning demonstrate strong transfer across datasets, from subsets to full corpora, and from smaller to larger models.

1 INTRODUCTION

The performance of modern AI systems is fundamentally shaped by the training data (Zha et al., 2025; Kumar et al., 2024; Zha et al., 2023; Whang et al., 2023). For large language models (LLMs), scaling requires assembling massive corpora from heterogeneous sources (Zhou et al., 2025; Weber et al., 2024; Langlais et al., 2026; Li et al., 2024a), which often introduces redundant and low-quality examples. Training data selection is hence an important component of LLM training. Many existing methods rank training examples by manually designed scores, such as entropy (Li et al., 2026), instruction-following difficulty (Li et al., 2024b), reducible training loss (Mindermann et al., 2022), and distance to target domain (Deng et al., 2026; Min et al., 2026), to name a few. These scoring rules lack the ability to dynamically adapt to new scenarios. Though each captures a potentially useful signal, it remains an open question how these scores may be combined to create more powerful ranking mechanisms.

Meta-learning for Training-data Selection (MTS) (Shen et al., 2025a; Pan et al., 2025; Ren et al., 2018; Wang et al., 2020; Du et al., 2026; Shu et al., 2019; Yu et al., 2025; Calian et al., 2026) provides a principled alternative, where the values of training data are learned directly from their effects on the validation loss $\mathcal{L}_{\text{val}}$. Let $\boldsymbol{\theta}$ denote the LLM parameters, $w_i > 0$ be the learnable weight for the loss of the i-th training data point, $\ell_i(\boldsymbol{\theta})$, and the vector $\boldsymbol{w} = (w_i)_i$ contain all w_i as components. MTS formulates data weighting as the following bilevel optimization problem:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathcal{L}_{\text{val}}\big(\boldsymbol{\theta}^*(\mathbf{w})\big), \qquad \text{s. t.} \quad \boldsymbol{\theta}^*(\mathbf{w}) = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{w}\|_1^{-1} \sum_i w_i \ell_i(\boldsymbol{\theta}). \tag{1}$$

This framework allows automatic identification of effective training data, which lead the model to generalize to the validation data.

Nevertheless, MTS in practice runs into two obstacles; the first is scalability. Computing the exact hypergradient $\nabla_{\boldsymbol{w}} \mathcal{L}_{\text{val}}(\boldsymbol{\theta}^*(\mathbf{w}))$ by unrolling the inner optimization is costly. Even unrolling one step and taking the finite-different approximation require three backward operations on $\boldsymbol{\theta}$ for one gradient step on $\boldsymbol{w}$ (Du et al., 2026; Shu et al., 2019), which is prohibitive expensive for large LLMs. Calian et al. (2026) manage to unroll 8 steps using 64 v5 TPUs, which is beyond the capability of most

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.

academic labs. With penalty-based reformulations (Shen & Chen, 2023; Xiao et al., 2023; Shen et al., 2025b), ScaleBiO (Pan et al., 2025) reduces the number of backward calls to two, whereas SEAL (Shen et al., 2025a) reduces it further to one, but incurs discernible performance losses.

The second challenge is generalization to unseen data. In the formulation of Ren et al. (2018); Shen et al. (2025a), every data point receives a separate weight w_i, and no generalization is possible. An alternative approach (Pan et al., 2025) is to assign a separate weight for each data domain (e.g., one dataset in a collection of multiple datasets), which provides generalization within each domain. However, this overlooks heterogeneity among data points in a single domain and cannot generalize to new data domains.

One intuitively appealing idea is to train a meta-network that outputs w_i for every data point. If the meta-network can generalize to unseen data points, we would be able to train on a small dataset and perform data selection on a much larger dataset, which would also substantially improve efficiency and scalability. Though the meta-network approach has been used by Shu et al. (2019); Du et al. (2026); Calian et al. (2026), no such generalization has been reported, not even for the extraordinarily large-scale experiments of Calian et al. (2026). Perhaps unsurprisingly, our experiments find simplistic training of the meta-network leads to unstable optimization (Figure 2) and poor generalization (Table 1 and 3).

Given the massive success of deep learning, the failure of the meta-network to generalize to unseen data is especially perplexing. Our analysis traces the failure to two properties of the training loss. First, as training progresses, most predicted data weights are suppressed toward zero. As a result, over time, the meta-network loses the ability to distinguish useful training data from harmful training data. Second, the loss causes the network to exclusively learn easy features and ignores hard feature, so the network becomes susceptible to shortcut learning.

To improve the scalability and generalization of MTS, we propose Transferable Example Scoring and Selection (TESS). At its core, TESS simplifies ScaleBiO (Pan et al., 2025) and introduces a simple mean squared error objective, which we call Pointwise Value Matching (PVM). TESS constructs sample-level pseudo-labels that capture the utility of training data points and uses these values as supervision for the data-selecting meta-network. Compared to the existing loss, the PVM loss is less focused on easy features, which curbs shortcut learning and promotes generalization. Further, TESS removes the requirement of ScaleBiO to simultaneously train two LLMs with the meta-network, cutting peak memory footprint by more than 20%. Experiments on LLM safety and targeted instruction tuning demonstrate substantially improved generalization of TESS across three settings: cross-dataset transfer, small-to-large model transfer, and subset-to-full-corpus transfer. Our contributions can be summarized as:

- We identify and theoretically analyze two limitations of applying the existing MTS training objective to a shared selection network: competitive weight suppression that undermines the identification of high-quality training data, and persistent preference for easy-to-learn features that can encourage shortcut learning.
- We propose TESS, a transferable data-selection framework that constructs sample-level pseudo-labels and uses them as supervision to learn a data selection network generalizable to unseen data. To the best of our knowledge, this is the first work that demonstrates generalization to unseen data in MTS.

2 Related Work

Meta-Learning Data Selection for LLMs. Scaling meta-learning data selection to LLMs requires efficiency in face of both huge parameter counts (Pan et al., 2025) and dataset sizes (Xie et al., 2023b). Traditional data selection methods (Ren et al., 2018; Shu et al., 2019) differentiate through model updates to compute the hypergradient, which incurs large time and space complexity for large models. ScaleBiO (Pan et al., 2025) and SEAL (Shen et al., 2025a) focus on scaling to model parameter counts. If data selection can be trained on a small dataset and generalize to unseen data, it would also significantly improve scalability to large datasets. However, such generalization remains a difficult challenge.

In terms of generalization, existing approaches can be broadly grouped into sample-level, domain-level, and meta-network methods. Example-level methods (Xie et al., 2023b; Xia et al., 2024; Shen

et al., 2025a; Calian et al., 2026) learn one weight for each training data point, thereby offering no generalization to new data points. Domain-level methods (Xie et al., 2023a; 2025; Pan et al., 2025; Yang et al., 2026; Wen et al., 2026) partition the training data into domains and assign one data weight to each domain. Chameleon (Xie et al., 2025) and Data Mixing Agent (Yang et al., 2026) further introduce a shared domain taxonomy that enables limited weight sharing among data domains. However, domain-level weights are too coarse to capture the crucial differences in data quality and content within each domain (Lee et al., 2024). Meta-network methods (Shu et al., 2019; Du et al., 2026; Calian et al., 2026) train a meta-network to output the weight for each data point directly. In theory, this should allow generalization to new data points; in practice, no generalization has been reported. Notably, DataRater (Calian et al., 2026) trains a separate scorer for each dataset, suggesting generalization across dataset is challenging even when the network are trained with a huge amount of data. With this paper, we analyze the reasons behind the challenge mathematically and offer a simple, practical, and high-performance solution.

Shortcut Learning. Shortcut learning occurs when models rely on easy features that are strongly predictive of training labels but whose correlations weaken or reverse under distribution shifts, impairing out-of-distribution generalization (Geirhos et al., 2020; Shah et al., 2020). Prior work distinguishes *predictivity*, how reliably a feature predicts training labels, from *availability*, how easily it can be extracted (Hermann et al., 2024). Scimeca et al. (2021) find that solutions favoring easy-to-learn cues tend to lie in flatter minima and are more abundant in parameter space, while Qiu et al. (2024) show that simpler or more strongly correlated spurious features can slow core-feature learning. We investigate these dynamics in data selection networks. We show that the ScaleBiO loss continues to reinforce easy-to-learn features that are susceptible to shortcut learning, while our proposed PVM promotes the learning of harder and more transferable features.

3 METHODOLOGY

3.1 PROBLEM FORMULATION

Let $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$ denote the training and target validation sets, with sizes N_{train} and N_{val}, respectively. We further use $\mathcal{D}_{\text{pool}}$ to denote the candidate pool to be scored. For example, $\mathcal{D}_{\text{pool}}$ can be a larger corpus containing $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{train}} \subsetneq \mathcal{D}_{\text{pool}}$, or a disjoint dataset, $\mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{pool}} = \varnothing$. Let $\ell_i(\boldsymbol{\theta})$ denote the token-averaged cross-entropy loss for the i-th data point under an LLM parameterized by $\boldsymbol{\theta}$. For the training examples, let $w_i > 0$ denote the weight of the i-th data point, $\boldsymbol{w} = (w_i)_{i=1}^{N_{\text{train}}}$ the weight vector, and $S = \|\boldsymbol{w}\|_1$ the total weight.

We aim to learn a data selection meta-network $s_{\boldsymbol{\phi}}$ that predicts sample weights and can score unseen examples in $\mathcal{D}_{\text{pool}}$. We parameterize $w_i = s_{\boldsymbol{\phi}}(x_i) > 0$ and denote the resulting weight vector by $\boldsymbol{w}(\boldsymbol{\phi})$. In MTS, learning $\boldsymbol{\phi}$ is formulated as bilevel optimization:

$$\boldsymbol{\phi}^* = \arg\min_{\boldsymbol{\phi}} \mathcal{L}\big(\boldsymbol{\theta}^*; \mathcal{D}_{\text{val}}\big) = \arg\min_{\boldsymbol{\phi}} N_{\text{val}}^{-1} \sum\nolimits_{i=1}^{N_{\text{val}}} \ell_i(\boldsymbol{\theta}^*), \tag{2}$$

$$\text{s.t.} \quad \boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}\big(\boldsymbol{\theta}, \boldsymbol{w}(\boldsymbol{\phi}); \mathcal{D}_{\text{train}}\big) = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{w}\|_1^{-1} \sum\nolimits_{i=1}^{N_{\text{train}}} w_i\, \ell_i(\boldsymbol{\theta}). \tag{3}$$

Here, $\mathcal{L}(\boldsymbol{\theta}; \mathcal{D})$ denotes the average per-sample loss, while $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{w}(\boldsymbol{\phi}); \mathcal{D})$ denotes the loss weighted by the meta-network $\boldsymbol{\phi}$. Under this formulation, we optimize $\boldsymbol{\phi}$ such that training the LLM with its predicted weights minimizes the loss on $\mathcal{D}_{\text{val}}$.

3.2 PRELIMINARIES ON SCALEBIO

ScaleBiO (Pan et al., 2025) provides an efficient framework for MTS at the LLM scale, optimizing data weights $\boldsymbol{w}$ for each data point. Let $\boldsymbol{\theta}_W$ denote the LLM parameters and $\boldsymbol{\theta}_U$ denote the parameters of an auxiliary LLM. Optimizing $\boldsymbol{\theta}_U$ on the training set yields $\boldsymbol{\theta}_U^* = \arg\min_{\boldsymbol{\theta}_U} \mathcal{L}(\boldsymbol{\theta}_U, \boldsymbol{w}; \mathcal{D}_{\text{train}})$. The original constraint in the bi-level problem (Eq. 1) can then rewritten as $\boldsymbol{\theta}_W = \boldsymbol{\theta}_U^*$.

Relaxing the constraint to a necessary condition that $\boldsymbol{\theta}_W$ and $\boldsymbol{\theta}_U^*$ have the same training loss, we obtain a new constraint: $\mathcal{L}(\boldsymbol{\theta}_W, \boldsymbol{w}; \mathcal{D}_{\text{train}}) - \min_{\boldsymbol{\theta}_U} \mathcal{L}(\boldsymbol{\theta}_U, \boldsymbol{w}; \mathcal{D}_{\text{train}}) = 0$. Building the Lagrangian function with the penalty coefficient $\alpha > 0$, we derive a new minimax problem:

$$\min_{\boldsymbol{w}, \boldsymbol{\theta}_W} \max_{\boldsymbol{\theta}_U} \quad \mathcal{L}(\boldsymbol{\theta}_W; \mathcal{D}_{\text{val}}) + \alpha \Big[\mathcal{L}(\boldsymbol{\theta}_W, \boldsymbol{w}; \mathcal{D}_{\text{train}}) - \mathcal{L}(\boldsymbol{\theta}_U, \boldsymbol{w}; \mathcal{D}_{\text{train}}) \Big]. \tag{4}$$

Algorithm 1 Overview of the TESS Framework

Require: Training set $\mathcal{D}_{\text{train}}$, validation set $\mathcal{D}_{\text{val}}$, candidate pool $\mathcal{D}_{\text{pool}}$, coefficient α

1: Obtain $\boldsymbol{\theta}_U^*$ by minimizing $\mathcal{L}(\boldsymbol{\theta}_U; \mathcal{D}_{\text{train}})$
2: Obtain $\boldsymbol{\theta}_W^*$ by minimizing $\mathcal{L}(\boldsymbol{\theta}_W; \mathcal{D}_{\text{val}}) + \alpha\mathcal{L}(\boldsymbol{\theta}_W; \mathcal{D}_{\text{train}})$
3: For each $x_i \in \mathcal{D}_{\text{train}}$, compute the pseudo-label $\Delta_i(\boldsymbol{\theta}_U^*, \boldsymbol{\theta}_W^*) \leftarrow \ell(\boldsymbol{\theta}_U^*; x_i) - \ell(\boldsymbol{\theta}_W^*; x_i)$
4: Train the selection network by optimizing $\boldsymbol{\phi}^* \leftarrow \arg\min_{\boldsymbol{\phi}} \mathcal{L}_{\boldsymbol{\phi}}^{\text{PVM}}(\boldsymbol{\phi}; \boldsymbol{\theta}_W^*, \boldsymbol{\theta}_U^*)$ (Eq.6)
5: For each $x_j \in \mathcal{D}_{\text{train}}$, predict $\hat{w}_j \leftarrow s_{\boldsymbol{\phi}^*}(x_j)$
6: **return** Predicted data values $\hat{\mathbf{w}}_{\text{pool}} = \{\hat{w}_j\}_{x_j \in \mathcal{D}_{\text{pool}}}$

To solve Eq.4, ScaleBiO simultaneously trains the two LLMs and data weights, resulting in substantial memory overhead. SEAL (Shen et al., 2025a) additionally eliminates the auxiliary LLM $\boldsymbol{\theta}_U$, but this comes at an accuracy penalty.

We write the ScaleBiO loss for data weights $\boldsymbol{w}$ explicitly:

$$\mathcal{L}_w^{\text{SBO}}(\boldsymbol{w}; \boldsymbol{\theta}_W, \boldsymbol{\theta}_U) = \|\boldsymbol{w}\|_1^{-1} \sum\nolimits_{i=1}^{N_{\text{train}}} w_i \big[\ell_i(\boldsymbol{\theta}_W) - \ell_i(\boldsymbol{\theta}_U)\big]. \tag{5}$$

It is evident that, to optimize L_w^{SBO}, we should assign high w_i to data points with a low loss on $\boldsymbol{\theta}_w$ and a high loss on $\boldsymbol{\theta}_U$. Note that $\boldsymbol{\theta}_w$ is trained on both the training set D_{train} and the validation set D_{val} but $\boldsymbol{\theta}_U$ is trained on D_{train} only. Therefore, ScaleBiO selects data points that benefit from additional training on D_{val}, which are likely more similar to D_{val} than D_{train}.

3.3 Transferable Example Scoring and Selection (TESS)

Incorporating a Selection Network. We extend ScaleBiO by replacing the independent data weights w_i with the data selection meta-network $s_{\boldsymbol{\phi}}(x_i)$. This yields the ScaleBiO meta-network objective (SBO):

$$\mathcal{L}_{\boldsymbol{\phi}}^{\text{SBO}}(\boldsymbol{\phi}; \boldsymbol{\theta}_W, \boldsymbol{\theta}_U) = \left(\sum\nolimits_{i=1}^{N_{\text{train}}} s_{\boldsymbol{\phi}}(x_i)\right)^{-1} \sum\nolimits_{i=1}^{N_{\text{train}}} s_{\boldsymbol{\phi}}(x_i) \big[\ell_i(\boldsymbol{\theta}_W) - \ell_i(\boldsymbol{\theta}_U)\big]. \tag{6}$$

However, empirically optimizing the meta-network with $\mathcal{L}_{\boldsymbol{\phi}}^{\text{SBO}}$ leads to unstable optimization (Figure 2) and poor generalization (Tables 1 and 3). We find this can be simply addressed by Pointwise Value Matching, presented below.

Pointwise Value Matching. The key idea is to use the loss difference between the two LLMs as a proxy for sample utility. Recall that $\boldsymbol{\theta}_U$ is trained only on $\mathcal{D}_{\text{train}}$, whereas $\boldsymbol{\theta}_W$ is additionally guided by $\mathcal{D}_{\text{val}}$. For each training example x_i, we define $\Delta_i(\boldsymbol{\theta}_U, \boldsymbol{\theta}_W) := \ell_i(\boldsymbol{\theta}_U) - \ell_i(\boldsymbol{\theta}_W)$. A larger Δ_i indicates a greater loss reduction under the validation-guided model relative to the training-only model, suggesting greater relevance to the target validation objective. We use this difference as a pseudo-label and train the meta-network to match it:

$$\mathcal{L}_{\boldsymbol{\phi}}^{\text{PVM}}(\boldsymbol{\phi}; \boldsymbol{\theta}_W, \boldsymbol{\theta}_U) = {N_{\text{train}}}^{-1} \sum\nolimits_{i=1}^{N_{\text{train}}} \left(s_{\boldsymbol{\phi}}(x_i) - \Delta_i(\boldsymbol{\theta}_U, \boldsymbol{\theta}_W)\right)^2. \tag{7}$$

Sequential Training of TESS. Unlike ScaleBiO, TESS separates the training of the three models. Specifically, using uniform sample weights, we obtain $\boldsymbol{\theta}_U^*$ by minimizing $\mathcal{L}(\boldsymbol{\theta}_U; \mathcal{D}_{\text{train}})$ and $\boldsymbol{\theta}_W^*$ by minimizing $\mathcal{L}(\boldsymbol{\theta}_W; \mathcal{D}_{\text{val}}) + \alpha\mathcal{L}(\boldsymbol{\theta}_W; \mathcal{D}_{\text{train}})$. We then freeze both models, compute $\Delta_i(\boldsymbol{\theta}_U^*, \boldsymbol{\theta}_W^*)$ for each training example, and train the meta-network by minimizing $\mathcal{L}_{\boldsymbol{\phi}}^{\text{PVM}}(\boldsymbol{\phi}; \boldsymbol{\theta}_W^*, \boldsymbol{\theta}_U^*)$. This sequential design has two advantages over ScaleBiO. First, it avoids simultaneously optimizing two LLMs and the selection network, reducing peak memory consumption. Second, it allows the pretrained LLMs and the randomly initialized meta-network to use independent optimization schedules. The trained meta-network directly scores unseen examples in $\mathcal{D}_{\text{pool}}$ without further optimization. The complete procedure of TESS is summarized in Algorithm 1.

4 Why Does the ScaleBiO Objective Perform Poorly?

We next examine why optimizing with the ScaleBiO objective can fail to produce a transferable meta-network. We identify two properties of the ScaleBiO objective: competitive weight suppression, which weakens sample discrimination, and a persistent preference for easy-to-learn features, which can encourage shortcut learning.

4.1 Data Weight Suppression and Loss of Discrimination

Our first observation is that the ScaleBiO objective drives most predicted weights toward zero, alongside losing the power of identifying good data (Figure 2). This occurs under both joint training in ScaleBiO and sequential training in TESS when training with $\mathcal{L}_{\phi}^{\mathrm{SBO}}$. Our analysis below strictly applies to sequential training in TESS, where the two LLMs are fixed. Empirically, we observe qualitatively similar dynamics under the joint training in ScaleBiO.

We first treat the weights of training examples as independent hyperparameters. For the selection network, $w_i = s_{\phi}(x_i)$ is updated through shared parameters ϕ. However, the update directions on w_i with and without ϕ are similar. With $\Delta_i^* := \Delta_i(\boldsymbol{\theta}_U^*, \boldsymbol{\theta}_W^*)$ and $N = N_{\mathrm{train}}$, the ScaleBiO objective and its gradient with respect to each data weight are

$$\mathcal{L}_{\phi}^{\mathrm{SBO}} = -\sum\nolimits_{i=1}^{N} p_i \Delta_i^*, \qquad \frac{\partial \mathcal{L}_{\phi}^{\mathrm{SBO}}}{\partial w_i} = -\frac{\Delta_i^* - \bar{\Delta}^*}{S}, \tag{8}$$

where $S = \sum_{j=1}^{N} w_j$ is the total weight, $p_i = w_i/S$ the normalized weight, and $\bar{\Delta}^* = \sum_i p_i \Delta_i^*$ the weighted-average pseudo-label. The derivation is provided in Appendix A.

The gradient reveals a competitive mechanism: examples are up-weighted when $\Delta_i^* > \bar{\Delta}^*(t)$ and down-weighted otherwise. As examples with larger pseudo-labels gain relative weight, the weighted mean rises, making it harder for other examples to receive positive updates. This can produce a *winner-takes-most* allocation, in which a single weight dominates the total mass. Given a lower bound $\delta_0 \geq 0$ of data weight, the following theorem is given by Du et al. (2026) (proof reproduced in Appendix B).

Theorem 1 (Concentration of normalized data weights p)**.** *Assume there is a unique index K such that $\Delta_K^* > \Delta_i^*$ for all $i \neq K$, then $\lim_{t\to\infty} p_K(t) = 1$ and $\lim_{t\to\infty} p_i(t) = 0$. For $\delta_0 = 0$, this limit is reached in finite time. For $\delta_0 > 0$, $1 - p_K(t) = O(t^{-1/3})$.*

Different from Du et al. (2026), who use weight concentration to analyze gradient signal-to-noise ratios, we examine its implications for sample discrimination. The ScaleBiO objective eventually allocates nearly all weight to one training data point. At the beginning, the meta-network can differentiate high-quality data points from low-quality data points by assigning them different weights. As training progresses, however, its outputs become less discriminative. This creates a strange phenomenon that more training leads to worsened data selection even on the training set itself.

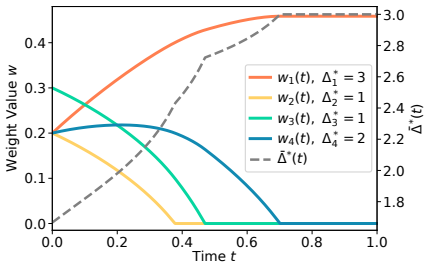

Figure 1: Simulation of the weight dynamics in Theorem 1. Curves show the weights $w_i(t)$, with fixed pseudo-labels Δ_i^* indicated in the legend. Initial weights are $(0.2, 0.2, 0.3, 0.2)$, and δ_0 is 10^{-6}.

Empirical verification. Figure 1 illustrates this behavior through a numerical simulation. The weight w_1, associated with the largest pseudo-label, becomes dominant, while the remaining weights reach the lower bound. Notably, w_4 initially increases but later decreases. Between $t = 0.4$ and $t = 0.6$, the weights can separate good data with the highest pseudo-labels, but this period is transient. Some data selection can be achieved by carefully tuning early stopping (consistent with Figure 9 of Calian et al. (2026)) but it requires significant trial and error.

To verify these dynamics, we construct a controlled dataset by mixing GSM8K and unsafe examples at a 2:1 ratio, together with a disjoint unseen set of the same composition. This way, we know exactly what data points are unsafe and should be discarded. We rank examples by predicted weights and evaluate how accurately the bottom one-third identifies unsafe examples on both seen and unseen data. Experimental details are provided in Appendix E. Figure 2 compares ScaleBiO with $\mathcal{L}_{\phi}^{\mathrm{SBO}}$, TESS with $\mathcal{L}_{\phi}^{\mathrm{SBO}}$, and TESS with $\mathcal{L}_{\phi}^{\mathrm{PVM}}$.

Under $\mathcal{L}_{\phi}^{\mathrm{SBO}}$, the 25th percentile, median, and 75th percentile of the predicted weights progressively approach zero, consistent with the competitive suppression analyzed above. The maximum weight initially increases but later decreases. This is expected because a shared selection network typically produces smoother outputs than a one-hot vector (Ma & Ying, 2021), causing even dominant weights to decrease as most outputs are suppressed.

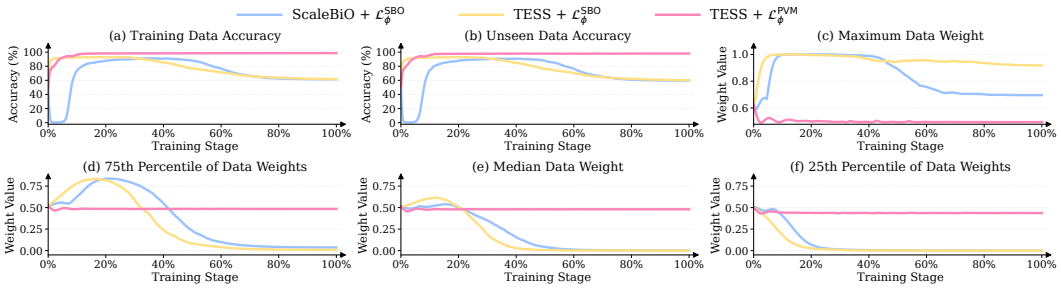

Figure 2: Training dynamics of ScaleBiO and TESS under different meta-network objectives. With SBO loss, most predicted weights shrink toward zero as accuracy declines. In contrast, PVM loss maintains nonvanishing weight quantiles and high accuracy.

Table 1: Safe–unsafe data classification accuracy under different shortcut strengths. Cov and ρ denote the empirical covariance and Pearson correlation coefficient between the shortcut feature and the label, respectively. Early, middle, and late stages correspond to 20%, 50%, and 80% of meta-network training, respectively. Acc_S and Acc_U denote seen and unseen accuracy, and $\text{Gap} = \text{Acc}_\text{S} - \text{Acc}_\text{U}$ denotes the generalization gap.

Shortcut (Cov / ρ)	Method	Early Stage			Middle Stage			Late Stage			Avg. Gap
		Acc_S	Acc_U	Gap	Acc_S	Acc_U	Gap	Acc_S	Acc_U	Gap	
Strong 0.17 / 0.57	TESS + $\mathcal{L}_\phi^{\text{PVM}}$	**98.84**	**79.38**	19.46	**99.56**	**84.62**	**14.94**	**99.56**	**84.97**	**14.59**	**16.33**
	TESS + $\mathcal{L}_\phi^{\text{SBO}}$	97.96	72.44	25.52	90.22	52.26	37.96	84.71	46.31	38.40	33.96
	ScaleBiO + $\mathcal{L}_\phi^{\text{SBO}}$	93.42	78.84	**14.58**	90.76	57.73	33.03	76.00	32.44	43.56	30.39
Weak 0.06 / 0.41	TESS + $\mathcal{L}_\phi^{\text{PVM}}$	**98.04**	**91.47**	**6.57**	**98.67**	**93.51**	**5.16**	**98.76**	**93.60**	**5.16**	**5.63**
	TESS + $\mathcal{L}_\phi^{\text{SBO}}$	96.53	80.71	15.82	87.82	65.78	22.04	84.00	60.53	23.47	20.44
	ScaleBiO + $\mathcal{L}_\phi^{\text{SBO}}$	91.64	82.40	9.24	90.13	71.73	18.40	75.56	55.56	20.00	15.88

Weight suppression accompanies declining discrimination. For ScaleBiO, seen and unseen accuracy first improve but eventually fall to around 60%. TESS with the same $\mathcal{L}_\phi^{\text{SBO}}$ exhibits a similar long-term decline, although it improves immediately at the beginning of training. We attribute this early difference to the quality of the supervision signal: ScaleBiO updates the meta-network while the two LLMs are still being optimized, whereas TESS computes loss gaps from well-trained, fixed LLMs. In contrast, TESS with $\mathcal{L}_\phi^{\text{PVM}}$ maintains stable, nonvanishing weight quantiles and reaches 98.58% seen accuracy and 97.96% unseen accuracy.

4.2 Easy-Feature Dominance and Shortcut Learning

The previous subsection links weight suppression to reduced sample discrimination. We now examine a second limitation: ScaleBiO can continue favoring easy-to-learn features after they have been substantially fitted, whereas PVM can shift learning toward harder features that explain the remaining prediction error. This distinction matters for generalization when an easy feature is predictive on the training data but unreliable on unseen data, thereby acting as a shortcut (Hermann et al., 2024; Geirhos et al., 2020).

Denote the empirical covariance as $\widehat{\text{Cov}}_N(u, v) := N^{-1} \sum_{i=1}^N (u_i - \bar{u})(v_i - \bar{v})$. Using the fixed pseudo-labels $\Delta_i^* = \Delta_i(\boldsymbol{\theta}_U^*, \boldsymbol{\theta}_W^*)$, for data features e h, we denote their covariance with pseudo-labels as

$$c_e = \widehat{\text{Cov}}_N(\Delta^*, e), \qquad c_h = \widehat{\text{Cov}}_N(\Delta^*, h). \tag{9}$$

We call e the *easy feature* and h the *hard feature* if $|c_e| > |c_h|$. For each training example x_i, let e_i and h_i be its easy and hard features, both standardized to zero mean and unit standard deviation. For simplicity, we analyze a one-layer meta-network with parameters $\boldsymbol{\phi} = (\beta_0, \beta_e, \beta_h)$ to predict weights for x_i. It predicts $w_i = s_{\boldsymbol{\phi}}(x_i) = \sigma(a_i)$, where $a_i = \beta_0 + \beta_e e_i + \beta_h h_i$ is the output logit, and σ is the sigmoid function.

To examine the learning preference for features, we compare the gap between gradients of easy and hard features in PVM and ScaleBiO training. We denote the feature gradients by $G_k^q(\phi) := \partial \mathcal{L}_{\boldsymbol{\phi}}^q / \partial \beta_k$, where $k \in \{e, h\}$ and $q \in \{\mathrm{SBO}, \mathrm{PVM}\}$.

Theorem 2 (Contrasting gradient-gap dynamics)**.** *At training step t, define gradient gap $D_t^q = |G_e^q(\phi_t)|^2 - |G_h^q(\phi_t)|^2$, then D_t^{PVM} decreases exponentially with t while D_t^{SBO} remains above its initial value. Specifically, there exists a $\lambda \in (0, 1)$ such that*

$$D_t^{\mathrm{PVM}} = (1-\lambda)^t D_0^{\mathrm{PVM}}, \qquad D_t^{\mathrm{SBO}} \geq D_0^{\mathrm{SBO}} \quad \forall\, t > 0 \tag{10}$$

The proof is provided in Appendix C. This comparison explains how PVM can gradually balance the information learned from easy and hard features, while ScaleBiO continues to prefer easy features. This links the training objective to shortcut learning. When the easy feature is predictive on the training data but unreliable on unseen data, ScaleBiO's persistent preference can reinforce reliance on this shortcut. In contrast, PVM mitigates the gradient gap between the easy feature and the hard feature. This provides a mechanism for TESS to move beyond training-specific shortcuts and improve data selection on unseen examples.

Empirical verification. To test shortcut reliance, we add shortcut features to the seen training data and flip them on unseen data (Appendix E). We use the gap between seen and unseen dataset accuracy as a proxy for generalization, with smaller gaps indicating better generalization on unseen data. Table 1 shows clear differences in training dynamics. Under both shortcut strengths, TESS with $\mathcal{L}_{\boldsymbol{\phi}}^{\mathrm{PVM}}$ improves unseen accuracy and reduces the gap from early to late training. In contrast, $\mathcal{L}_{\boldsymbol{\phi}}^{\mathrm{SBO}}$ leads to lower unseen accuracy and larger gaps over training in both frameworks. These results are consistent with our analysis that SBO persistently learning more shortcut features, and support PVM as a more effective objective for learning transferable meta-networks. Moreover, all variants exhibit larger average gaps under strong shortcuts, indicating that stronger spurious correlations make generalization more difficult.

5 Experiments

5.1 LLM Safety

We evaluate whether a meta-network can identify training examples that increase harmful behavior finetuning a target LLM. We use Alpaca (Taori et al., 2023; Peng et al., 2023) and Dolly (Conover et al., 2023) as candidate pools $\mathcal{D}_{\mathrm{pool}}$. The validation set from He et al. (2024) contains 100 harmful instructions paired with safe refusal responses. We finetune the target LLM on 1,000 lowest-scoring examples and evaluate it on DirectHarm4 (DH4) (Lyu et al., 2024), HarmBench (HB) (Mazeika et al., 2024), and HEx-PHI (HEx) (Qi et al., 2024). We use LlamaGuard 3 (Llama Team, 2024) as the classifier to compute attack success rate (ASR) from generated responses; higher ASR indicates more effective harmful-data selection. The target models are Llama-3-8B-Instruct, Llama-3.1-8B-Instruct (Llama Team, 2024), and Qwen2.5-7B-Instruct (Qwen et al., 2025). Implementation details are provided in Appendix D.

Settings and Baselines. We consider two settings. In *Training*, the meta-network is trained and applied on the same candidate dataset, with $\mathcal{D}_{\mathrm{train}} = \mathcal{D}_{\mathrm{pool}}$. In *Generalization*, it is trained on Alpaca and directly applied to Dolly, or vice versa, with $\mathcal{D}_{\mathrm{train}} \cap \mathcal{D}_{\mathrm{pool}} = \varnothing$. We compare against the random selection baseline and two groups of task-agnostic methods. GradSafe (Xie et al., 2024) and Bi-Anchoring (Bi-Anchor) (He et al., 2024) are heuristic methods that rank examples using distances relative to the validation set. SEAL (Shen et al., 2025a) and ScaleBiO (Pan et al., 2025) are meta-learning methods that learn data weights. We create two variants for each. SEAL$_w$ and SBO$_w$ optimize one weight per example, whereas SEAL$_\phi$ and SBO$_\phi$ train a meta-network $s_{\boldsymbol{\phi}}$ to predict weights.

Main Results. Table 2 shows the results. Under *Training* settings, TESS achieves the strongest performance, exceeding the strongest learning baseline SBO$_w$ by 16.27% and the strongest heuristic Bi-Anchor by 11.04%. Although Bi-Anchor remains stronger on Llama3-Alpaca, TESS achieves the best in 14 of the 18 comparisons. These gains stem from the optimization stability of TESS. In contrast, simply replacing individual weights with a shared meta-network does not consistently help: SEAL$\boldsymbol{\phi}$ and SBO$_\phi$ underperform their weight-based counterparts by 1.50 and 4.68 points on average. This suggests that the training objective is critical for effective selection.

Table 2: Attack success rate (ASR, %; higher is better) for harmful-data selection on Alpaca and Dolly. Best and second-best results are **bold** and underlined, ranked separately within two settings.

Setting		Training								Generalization		
Dataset	Bench	Random	Task-agnostic Heuristics		Learnable Weighting							
			GradSafe	Bi-Anchor	SEAL_w	SBO_w	SEAL_ϕ	SBO_ϕ	TESS	SEAL_ϕ	SBO_ϕ	TESS
Target LLM: Llama3-8B-Instruct												
Alpaca	DH4	25.00	28.00	**49.00**	26.75	38.25	12.50	12.50	36.75	8.50	6.50	**35.50**
	HB	15.00	16.00	**35.00**	13.50	21.00	9.00	7.00	20.50	6.00	4.00	**25.00**
	HEx	6.55	8.97	**24.58**	6.90	10.69	5.86	3.44	11.38	3.10	3.44	**15.52**
Dolly	DH4	55.25	75.00	74.50	60.75	69.25	73.00	74.50	**82.25**	56.25	43.00	**71.25**
	HB	39.25	82.00	75.00	59.50	73.00	71.25	69.50	**85.50**	43.50	27.50	**59.50**
	HEx	31.38	74.14	67.59	42.76	53.44	61.03	63.44	**81.72**	30.69	22.07	**52.41**
Average		28.74	47.35	**54.28**	35.03	44.27	38.77	38.40	53.02	24.67	17.75	**43.20**
Target LLM: Llama3.1-8B-Instruct												
Alpaca	DH4	22.50	7.50	11.00	27.75	20.50	8.50	5.50	**40.25**	2.25	11.00	**34.25**
	HB	**18.50**	5.00	12.50	13.00	8.00	7.50	3.50	**18.50**	2.50	7.00	**16.50**
	HEx	8.97	3.45	3.10	6.90	2.41	2.76	1.03	**12.41**	1.03	4.83	**9.66**
Dolly	DH4	54.00	59.50	67.25	71.75	63.75	64.00	74.50	**81.00**	71.50	60.25	**80.25**
	HB	51.00	60.50	50.50	65.00	58.50	55.00	60.00	**80.00**	69.00	50.50	**81.00**
	HEx	29.30	33.79	40.00	38.62	33.10	33.45	42.07	**60.34**	40.69	28.28	**71.38**
Average		30.71	28.29	30.73	37.17	31.04	28.54	31.10	**48.75**	31.16	26.98	**48.84**
Target LLM: Qwen2.5-7B-Instruct												
Alpaca	DH4	27.50	26.00	**44.50**	20.00	27.00	28.25	11.75	**44.50**	8.75	11.00	**38.75**
	HB	11.00	10.00	**24.50**	9.00	8.00	17.50	3.50	23.50	3.00	2.50	**16.00**
	HEx	13.10	12.07	24.80	6.55	8.28	9.31	6.56	**24.83**	7.24	6.56	**18.28**
Dolly	DH4	50.50	66.50	60.50	49.75	65.75	60.25	59.75	**86.50**	48.00	62.25	**84.50**
	HB	36.00	60.00	60.50	65.50	61.50	43.00	45.50	**87.00**	34.00	50.50	**83.50**
	HEx	32.41	51.03	42.07	51.03	50.34	45.86	44.48	**88.62**	27.59	29.66	**80.00**
Average		28.42	37.60	42.81	33.64	36.81	34.03	28.59	**59.16**	21.43	27.08	**53.51**
Overall Average		29.29	37.75	42.61	35.28	37.38	33.78	32.70	**53.64**	25.76	23.94	**48.51**

Under the *Generalization* setting, TESS outperforms SEAL_ϕ and SBO_ϕ in all 18 evaluations, surpassing SEAL_ϕ and SBO_ϕ by 22.76 and 24.58 points in average ASR, respectively, while dropping only 5.13% from its training performance. These results support our analysis that PVM learns more transferable selection signals rather than relying on training-specific shortcuts.

Small-to-Large Model Transfer. We further evaluate whether a meta-network trained using a smaller LLM can select useful data for a larger model. We train the meta-network with Qwen2.5-0.5B-Instruct and use it to select data for fine-tuning Qwen2.5-7B-Instruct. As shown in Table 5 (Appendix F), this reduces training time by 5.65–$6.60\times$ while retaining 77.53% and 83.67% of the corresponding large-model performance under the *Training* and *Generalization* settings. This demonstrates that TESS can transfer selection signals across model scales with substantially lower training cost. This offers a promising direction for scaling data selection to larger models efficiently.

Ablation Study. We ablate the training losses within the TESS framework on Qwen2.5-7B-Instruct, comparing PVM Loss (*defalt*), SBO Loss, KL Loss, together with Pseudo-Label Ranking, which directly ranks examples by Δ^* without training a meta-network. KL Loss (Appendix D.5) matches normalized meta-network outputs to a pseudo-label-reweighted target distribution that favors examples with higher estimated utility.

Under the *Training* setting, TESS with PVM Loss achieves the highest ASR across all six dataset–benchmark pairs, outperforming SBO Loss and KL Loss by 21.60% and 16.24%, respectively. These gains stem from the more stable optimization of PVM. Notably, PVM Loss also outperforms *Pseudo-Label Ranking* by 18.15%. One possible explanation is that the meta-network may implicitly denoise pseudo-labels by learning shared patterns (Liu et al., 2020; Li et al., 2020). Under the *Generalization* setting, PVM Loss exceeds SBO Loss and KL Loss by 24.61% and 25.07% on average, respectively. Moreover, its average ASR decreases by only 5.65 points relative to the *Training* setting, compared with drops of 8.67% for SBO Loss and 14.49% for KL Loss. The stronger

Table 3: Ablation of Different Training Losses for the Meta-network on Qwen2.5-7B-Instruct.

Setting	Method	Alpaca			Dolly			Avg.
		DH4	HB	HEx	DH4	HB	HEx	
Training	TESS w/ PVM Loss	**44.50**	**23.50**	**24.83**	**86.50**	**87.00**	**88.62**	**59.16**
	TESS w/ SBO Loss	39.50	19.50	22.06	58.25	40.50	45.52	$37.56_{-21.60}$
	TESS w/ KL Loss	40.75	19.00	23.86	60.75	53.50	59.66	$42.92_{-16.24}$
	Pseudo-Label Ranking	11.25	7.50	5.86	75.75	75.00	70.69	$41.01_{-18.15}$
Generalization	TESS w/ PVM Loss	**38.75**	**16.00**	**18.28**	**84.50**	**83.50**	**80.00**	**53.51**
	TESS w/ SBO Loss	12.00	9.50	7.24	59.25	44.00	41.37	$28.89_{-24.61}$
	TESS w/ KL Loss	11.75	4.00	6.55	61.00	43.50	43.79	$28.43_{-25.07}$

Table 4: Llama-2-7B performance after fine-tuning on N selected Tulu V2 examples.

Method	GSM8K				CodeX				Overall Avg.
	N=1K	N=5K	N=10K	Avg.	N=5K	N=10K	N=20K	Avg.	
Random	13.41	15.92	16.13	15.15	29.05	27.03	27.70	27.93	21.54
LESS	18.80	22.36	22.59	21.25	25.00	26.35	28.37	26.57	$23.91_{+2.37}$
RDS+	17.43	21.60	23.57	20.87	28.37	26.35	**29.73**	28.15	$24.51_{+2.97}$
SEAL_w	14.78	16.37	17.89	16.35	24.32	25.00	25.68	25.00	$20.67_{-0.87}$
SBO_w	15.01	17.74	18.65	17.13	26.35	25.68	27.03	26.35	$21.74_{+0.20}$
SEAL_ϕ	17.74	19.56	19.48	18.93	24.32	27.03	26.35	25.90	$22.41_{+0.87}$
SBO_ϕ	17.43	18.65	19.03	18.37	26.35	25.68	27.70	26.58	$22.47_{+0.93}$
Ours	**20.43**	**25.47**	**25.70**	**23.87**	**29.73**	**28.37**	29.05	**29.05**	$\mathbf{26.46}_{+4.92}$

unseen-data performance and smaller degradation indicate that PVM learns a more transferable meta-network.

5.2 Targeted Instruction Tuning

We evaluate if TESS trained on a small subset can generalize to the full instruction tuning corpus. GSM8K (Cobbe et al., 2021) and Codex (Chen et al., 2021) are the target tasks. We train the meta-network using a random 50K-example subset of Tulu V2 (Ivison et al., 2025) and a task-specific validation set, then apply it to select training data from the 197K full Tulu V2. Following Nayak et al. (2026), we finetune Llama-2-7B on the selected examples and report GSM8K accuracy and Codex pass@10. Implementation details are provided in Appendix D. We compare against random selection, heuristic-based methods, and learnable-weighting baselines. The heuristic methods, LESS (Xia et al., 2024) and RDS+ (Ivison et al., 2025), rank candidate examples using predefined measures of relevance to the target validation set. All four learnable baselines are trained on the same 50K subset. SEAL_w and SBO_w are restricted to select from their training set.

Main Results. Table 4 shows that TESS achieves the best average performance on both tasks and ranks first in five of six settings, exceeding Random by 4.92 points overall. TESS improves over heuristic-based selection, exceeding LESS and RDS+ in overall average by 2.55% and 1.95%, respectively. The advantage is more pronounced on GSM8K, where TESS outperforms the strongest heuristic baseline, LESS, in task-average performance by 2.62%. TESS also consistently outperforms all learnable-weighting baselines. Compared with SEALϕ and SBOϕ, which use the same 50K training subset and full selection pool, TESS improves the overall average by 4.05 and 3.99 points, respectively. This controlled comparison shows that the performance gains come from the new loss rather than merely using a shared meta-network.

Regarding training efficiency, SEAL_w and SBO_w optimize sample-specific weights and thus must be trained on the full 200K pool, whereas TESS trains on only 50K examples and generalizes to the full pool. As a result, SBO_w and SEAL_w require 59.8 and 39.6 hours, respectively, corresponding to 3.46× and 1.95× more training time than the 13.4 hours required by TESS. SBO_w also incurs higher memory usage due to jointly training multiple models, requiring 188.2 GB compared with 156.8 GB for TESS, an increase of 31.4,GB (20.03%).

6 Conclusion

We introduced TESS, a transferable data-selection framework that learns a selector from sample-level pseudo-labels. We identify weight suppression and easy-feature preference of ScaleBiO objective and address them with PVM. TESS shows strong transfer across datasets, corpus scales, and model sizes.

AI USE STATEMENT

In this work, we used generative AI tools to formulate mathematical claims, assist in the writing of proofs, provide feedback on research methodology or experiments, help develop theoretical models or conceptual frameworks, provide critical ingredients for proving mathematical claims, and assist with translation. We have not used generative AI tools to generate synthetic data sets, implement methods, support qualitative and thematic data analysis, or interpret results, and propose or refine hypotheses and clean and reformat dataset are not applicable to this work.

Additionally, we used generative AI tools to edit software code, search for information, and edit a research paper to improve readability. We have not used generative AI tools to create or modify scientific figures or images, suggest experimental parameters, draft parts of a research paper, summarize or analyse existing literature, discover research topics or identify gaps, brainstorm, identify relevant literature, format references, suggest a structure for a research paper, or propose a title or keywords for a research paper. Formulate questions for surveys or interviews, creation of artifacts, and transcribe recordings of research material are not applicable to this work.

We have reviewed all AI-assisted work. All AI-assisted text, mathematical results, and information-search results were manually verified and revised by the authors as needed. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the models, datasets, training configurations, and evaluation protocols in Section 5.1 and Appendix D. All datasets and pretrained models used in this work are from prior work and are publicly available. We will also release our code upon acceptance to facilitate reproducibility.

REFERENCES

Dan Andrei Calian, Greg Farquhar, Iurii Kemaev, Luisa Zintgraf, Matteo Hessel, Jeremy Shar, Junhyuk Oh, András György, Tom Schaul, Jeff Dean, et al. Datarater: Meta-learned dataset curation. *Advances in Neural Information Processing Systems*, 38:20060–20092, 2026.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. doi: 10.48550/arXiv.2107.03374. URL https://arxiv.org/abs/2107.03374.

Ruibo Chen, Yihan Wu, Lichang Chen, Guodong Liu, Qi He, Tianyi Xiong, Chenxi Liu, Junfeng Guo, and Heng Huang. Your vision-language model itself is a strong filter: Towards high-quality instruction tuning with data selection. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4156–4172, 2024. doi: 10.18653/v1/2024.findings-acl.246.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. doi: 10.48550/arXiv.2110.14168. URL https://arxiv.org/abs/2110.14168.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

Xun Deng, Han Zhong, Rui Ai, Fuli Feng, Zheng Wang, and Xiangnan He. Less is more: Improving llm alignment via preference data selection. *Advances in Neural Information Processing Systems*, 38:161259–161285, 2026.

Zilin Du, Junqi Zhao, and Boyang Albert Li. On the difficulty of learning a meta-network for training data selection. *arXiv preprint arXiv:2606.00571*, 2026.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Luxi He, Mengzhou Xia, and Peter Henderson. What is in your safe data? identifying benign data that breaks safety. *arXiv preprint arXiv:2404.01099*, 2024.

Katherine Hermann, Hossein Mobahi, Thomas Fel, and Michael Mozer. On the foundations of shortcut learning. In *International Conference on Learning Representations*, volume 2024, pp. 43832–43868, 2024.

Hamish Ivison, Muru Zhang, Faeze Brahman, Pang Wei Koh, and Pradeep Dasigi. Large-scale data selection for instruction tuning. *arXiv preprint arXiv:2503.01807*, 2025.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704, 2023.

Sushant Kumar, Sumit Datta, Vishakha Singh, Sanjay Kumar Singh, and Ritesh Sharma. Opportunities and challenges in data-centric ai. *IEEE Access*, 12:33173–33189, 2024.

Pierre-Carl Langlais, Pavel Chizhov, Catherine Arnett, Carlos Hinostroza, Mattia Nee, Eliot Jones, Irène Girard, David Mach, Anastasia Stasenko, and Ivan Yamshchikov. Common corpus: The largest collection of ethical data for llm pre-training. In *International Conference on Learning Representations*, volume 2026, pp. 25827–25860, 2026.

Jaewoo Lee, Boyang Li, and Sung Ju Hwang. Concept-skill transferability-based data selection for large vision-language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5060–5080, 2024. doi: 10.18653/v1/2024.emnlp-main.291.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024a.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7602–7635, 2024b.

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International conference on artificial intelligence and statistics*, pp. 4313–4324. PMLR, 2020.

Xiaoyuan Li, Yubo Ma, Chengpeng Li, Fengbin Zhu, Yiyao Yu, Keqin Bao, Wenjie Wang, Fuli Feng, and Dayiheng Liu. Unified data selection for llm reasoning. *arXiv preprint arXiv:2605.22389*, 2026.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33:20331–20342, 2020.

AI @ Meta Llama Team. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

Kaifeng Lyu, Haoyu Zhao, Xinran Gu, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. Keeping llms aligned after fine-tuning: The crucial role of prompt templates. *Advances in Neural Information Processing Systems*, 37:118603–118631, 2024.

Chao Ma and Lexing Ying. On linear stability of sgd and input-smoothness of neural networks. *Advances in Neural Information Processing Systems*, 34:16805–16817, 2021.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

Guanghui Min, Tianhao Huang, Ke Wan, and Chen Chen. Gist: Targeted data selection for instruction tuning via coupled optimization geometry. *arXiv preprint arXiv:2602.18584*, 2026.

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, et al. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, pp. 15630–15649. PMLR, 2022.

Nihal V. Nayak, Paula Rodriguez-Diaz, Neha Hulkund, Sara Beery, and David Alvarez-Melis. A critical look at targeted instruction selection: Disentangling what matters (and what doesn't). In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.14696.

Rui Pan, Xiang Liu, Shizhe Diao, Renjie Pi, Jipeng Zhang, Chi Han, and Tong Zhang. Lisa: Layerwise importance sampling for memory-efficient large language model fine-tuning. *Advances in Neural Information Processing Systems*, 37:57018–57049, 2024.

Rui Pan, Dylan Zhang, Hanning Zhang, Xingyuan Pan, Minrui Xu, Jipeng Zhang, Renjie Pi, Xiaoyu Wang, and Tong Zhang. Scalebio: Scalable bilevel optimization for llm data reweighting. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31959–31982, 2025.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, volume 2024, pp. 30988–31043, 2024.

GuanWen Qiu, Da Kuang, and Surbhi Goel. Complexity matters: Feature learning in the presence of spurious correlations. In *Forty-first International Conference on Machine Learning*, 2024.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report, January 2025. URL http://arxiv.org/abs/2412.15115. arXiv:2412.15115 [cs].

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pp. 4334–4343. PMLR, 2018.

Luca Scimeca, Seong Joon Oh, Sanghyuk Chun, Michael Poli, and Sangdoo Yun. Which shortcut cues will dnns choose? a study from the parameter-space perspective. *arXiv preprint arXiv:2110.03095*, 2021.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Han Shen and Tianyi Chen. On penalty-based bilevel gradient descent method. In *International conference on machine learning*, pp. 30992–31015. PMLR, 2023.

Han Shen, Pin-Yu Chen, Payel Das, and Tianyi Chen. Seal: Safety-enhanced aligned llm fine-tuning via bilevel data selection. In *International Conference on Learning Representations*, volume 2025, pp. 31243–31264, 2025a.

Han Shen, Zhuoran Yang, and Tianyi Chen. Principled penalty-based methods for bilevel reinforcement learning and rlhf. *Journal of Machine Learning Research*, 26(114):1–49, 2025b.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32, 2019.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

Yulin Wang, Jiayi Guo, Shiji Song, and Gao Huang. Meta-semi: A meta-learning approach for semi-supervised learning. *arXiv preprint arXiv:2007.02394*, 2020.

Maurice Weber, Daniel Y Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, et al. Redpajama: an open dataset for training large language models. *Advances in neural information processing systems*, 37: 116462–116492, 2024.

Bingbing Wen, Sirajul Salekin, Feiyang Kang, Bill Howe, Lucy Lu Wang, Javier Movellan, and Manjot Bilkhu. MixAtlas: Uncertainty-aware data mixture optimization for multimodal llm midtraining, 2026.

Steven Euijong Whang, Yuji Roh, Hwanjun Song, and Jae-Gil Lee. Data collection and quality challenges in deep learning: a data-centric ai perspective: Se whang et al. *The VLDB Journal*, 32 (4):791–813, 2023.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning, 2024. URL `https://arxiv.org/abs/2402.04333`.

Quan Xiao, Songtao Lu, and Tianyi Chen. A generalized alternating method for bilevel learning under the polyak–łojasiewicz condition. *arXiv preprint arXiv:2306.02422*, 2023.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining. In *Advances in Neural Information Processing Systems*, volume 36, 2023a. doi: 10.52202/075280-3059.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S. Liang. Data selection for language models via importance resampling. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34201–34227, 2023b. doi: 10.52202/075280-1482.

Wanyun Xie, Francesco Tonin, and Volkan Cevher. Chameleon: A flexible data-mixing framework for language model pretraining and finetuning, 2025.

Yueqi Xie, Minghong Fang, Renjie Pi, and Neil Gong. Gradsafe: Detecting jailbreak prompts for llms via safety-critical gradient analysis. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 507–518, 2024.

Kailai Yang, Xiao Liu, Lei Ji, Hao Li, Xiao Liang, Zhiwei Liu, Yeyun Gong, Peng Cheng, and Mao Yang. Data mixing agent: Learning to re-weight domains for continual pre-training, 2026.

Yang Yu, Kai Han, Hang Zhou, Yehui Tang, Kaiqi Huang, Yunhe Wang, and Dacheng Tao. Llm data selection and utilization via dynamic bi-level optimization. *arXiv preprint arXiv:2507.16178*, 2025.

Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, and Xia Hu. Data-centric ai: Perspectives and challenges. *arXiv preprint arXiv:2301.04819*, 2023.

Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. Data-centric artificial intelligence: A survey. *ACM Computing Surveys*, 57(5):1–42, 2025.

Xuanhe Zhou, Junxuan He, Wei Zhou, Haodong Chen, Zirui Tang, Haoyu Zhao, Xin Tong, Guoliang Li, Youmin Chen, Jun Zhou, et al. A survey of LLM $\times$ DATA. *arXiv preprint arXiv:2505.18458*, 2025.