A Review on Data Selection for LLM Finetuning

Boyi Liu

Affiliations: City University of Hong Kong, Beihang University

Data selection has become a critical component of the large language model (LLM) training pipeline: across pretraining, supervised fine-tuning (SFT), and preference alignment (e.g., DPO), choosing the most valuable subset from a large, noisy, and quality-heterogeneous candidate pool often improves downstream performance more than simply scaling up data, while substantially reducing compute cost. However, existing methods differ widely in their scoring signals, comparison baselines, scoring models, selection algorithms, and execution timing, and a unified framework for comparison is lacking. This survey proposes a five-dimensional taxonomy that decomposes data selection into: ① the scoring metric, ② the comparison target, ③ the scoring model, ④ the selection policy, and ⑤ the selection timing (offline vs. online). Taking representative works as examples, we characterize each directly along these five dimensions and analyze their design trade-offs. Along the way we surface a third scoring-model regime beyond proxy and self—the external LLM / reward-model judge (and, at the extreme, human curation)—and show that quality-oriented selection is almost always paired with an explicit diversity or coverage objective. We close by summarizing the core open challenges and outlining future directions.

Main Contact: boy.liu@my.cityu.edu.hk
Github: https://github.com/boyi-liu/Awesome-Data-Selection-for-LLM-Finetuning
Last Updated: June 24, 2026

1 Introduction

For much of the recent history of large language models (LLMs), progress has been driven by a simple recipe: more parameters, more data, and more compute. Yet as models approach the practical limits of available data and compute, a growing body of evidence indicates that the *quality* and *composition* of training data matter at least as much as its sheer *quantity*. Works such as LIMA Zhou et al. (2023) and Alpaca-style instruction tuning have shown that a small, carefully curated set of high-quality examples can match or exceed the alignment quality obtained from vastly larger but noisier corpora. Conversely, redundant, low-quality, or off-distribution samples not only waste compute but can actively degrade performance.

This realization has crystallized into an active research direction: *data selection*. Given a large pool of candidate training examples, the goal is to choose a subset that maximizes downstream performance under a fixed data or compute budget. Data selection spans the entire LLM training lifecycle—corpus filtering and deduplication during pretraining, example selection during supervised fine-tuning (SFT), and preference-pair curation during alignment (e.g., DPO). Across all of these stages the central questions are the same: *what makes a training example valuable, and how do we find such examples efficiently?*

1.1 Motivation

Although data selection methods have proliferated, they differ wildly in their underlying motivations and technical machinery. Some define a sample's value by whether "the model can currently learn from it" (loss- or difficulty-based signals); others measure a sample's gradient contribution to a target objective (influence-based signals); still others focus on a sample's fit to the model's distribution or on the diversity of the selected subset. Some methods rely on an external proxy model to score data offline; others use the model being trained itself, scoring data online within the training loop. These methods are typically presented in isolation, and the literature lacks a unified coordinate system in which their essential similarities and differences can be

compared.

The central contribution of this survey is precisely such a framework. We argue that any data selection method, however intricate on the surface, can be cleanly characterized along *five orthogonal dimensions*. Equipped with this framework, a reader can quickly locate a new method in the design space, understand its trade-offs, and identify combinations that remain underexplored.

1.2 A Five-Dimensional Taxonomy

We decompose LLM data selection into the following five dimensions, which structure the remainder of this survey:

1. **Scoring metric** (Section 3): *What signal measures a sample's value?* Loss/difficulty, gradient/influence, distribution-fit/diversity, or preference margin?
2. **Comparison target** (Section 4): *Relative to what is the score computed?* The sample/model itself (self-comparison), or an external anchor (a validation set, anchor data, or a target distribution)?
3. **Scoring model** (Section 5): *Which model produces the score?* A separate proxy model, or the model being trained itself (self)?
4. **Selection policy** (Section 6): *Given scores, how is data chosen?* Hard filtering (top-k/threshold), soft reweighting, or greedy/bandit/search?
5. **Selection timing** (Section 7): *When does selection happen?* Once before training (offline), or dynamically during training (online)?

These dimensions are not perfectly independent—self-scoring naturally favors online selection, while proxy-scoring leans offline—but each captures a distinct degree of freedom, and together they suffice to locate the vast majority of existing methods.

1.3 Representative Works and Overview

To ground the analysis, we study representative works spanning pretraining-corpus filtering, supervised fine-tuning, and preference alignment. Rather than sorting them into fixed families, we characterize each work *directly along the five dimensions*: Table 1 maps every method onto its ① metric, ② comparison target, ③ scoring model, ④ selection policy, and ⑤ timing, and serves as the master index for the analysis in Sections 3–7. Two observations recur across the set and shape the discussion that follows: the scoring model (Dimension ③) is not a binary proxy-vs-self choice but also admits an external-LLM / reward-model judge and a human-curation extreme; and quality-oriented selection is almost always paired with an explicit *diversity / coverage* objective.

Table 1 A five-dimensional taxonomy of the representative data selection methods surveyed. The five dimensions are ① scoring metric, ② comparison target, ③ scoring model, ④ selection policy, and ⑤ timing. Each cell leads with the subcategory (Sections 3–7); parentheses give the method-specific detail.

Method	① Metric	② Comparison	③ Model	④ Policy	⑤ Timing
IFD / Cherry-LLM Li et al. (2024b)	Loss-Based (instruction-following difficulty; loss ratio w/ vs. w/o instruction)	Self-Comparison (w/ vs. w/o instruction)	Self Model (warmed-up)	Hard Filtering (mid-IFD samples)	Offline
GRAPE Zhang et al. (2025a)	Representation-Based (response likelihood under base model; distribution fit)	Self-Comparison (candidate responses)	Self Model (base model)	Hard Filtering (top-prob. response)	Offline
MIWV Jiang et al. (2025)	Loss-Based (response gap w/ vs. w/o ICL; model "weakness")	Self-Comparison (w/ vs. w/o ICL)	Self Model	Hard Filtering (high-MIWV samples)	Offline
Priority-on-High-Quality Zhang et al. (2025b)	Representation-Based (noise-injection consistency + diversity)	Self-Comparison (pre-/post-noise)	Self Model	Hard Filtering (high-consistency + diverse)	Offline
Selective DPO Gao et al. (2025)	Loss-Based (pair difficulty; ref. validation loss)	Anchor Comparison (reference-model capacity threshold)	Self Model (+ reference)	Hard Filtering (drop too-hard pairs)	Offline
Less-is-More Deng et al. (2025)	Preference Margin (ext. + implicit DPO reward margin)	Anchor Comparison (external reward estimator)	Judge Model (external RM + self)	Hard Filtering (max-margin)	Offline

continued on next page

Table 1 (continued)

Method	① Metric	② Comparison	③ Model	④ Policy	⑤ Timing
LESS Xia et al. (2024)	Gradient-Based (influence; low-rank, Adam-aware grad. sim.)	Anchor Comparison (target val. gradients)	Proxy Model (small LoRA)	Hard Filtering (top-k by influence)	Offline
Instruction Mining Cao et al. (2023)	Loss-Based (predicted loss; linear quality rule)	Anchor Comparison (predicted quality scale)	Proxy Model (+ reward model)	Optimization (BlendSearch)	Offline
GOT-D Kang et al. (2024)	Gradient-Based (optimal-transport gradient)	Anchor Comparison (target vs. pretrain dist.)	Proxy Model	Hard Filtering (most dist.-shifting)	Offline
CLUES Zhao et al. (2024)	Gradient-Based (grad. inner-product w/ anchor; accumulated)	Anchor Comparison (high-quality anchor)	Proxy Model (federated local models + anchor)	Hard Filtering (match anchor dynamics)	Offline
Compute-Constrained Yin and Rush (2024)	Utility-per-compute (meta-analysis; no single metric)	Utility vs. cost (budget trade-off)	Meta-analysis (no single scoring model)	Optimization (optimal under budget)	Offline
CADS Wan et al. (2025)	Gradient-Based (importance from within-budget eval)	Anchor Comparison (budget-trained inner-model eval)	Proxy Model (budget inner)	Optimization (bilevel)	Offline
GREATS Wang et al. (2024)	Gradient-Based (Taylor-approx. val.-loss drop; ghost dot-product)	Anchor Comparison (held-out val. set)	Self Model	Optimization (greedy batch selection)	Online
LEAD Lin et al. (2025)	Loss-Based (dynamic uncertainty; loss + grad change + EMA)	Self-Comparison (own loss dynamics)	Self Model (in-loop)	Optimization (bandit cluster + IDU)	Online
UDS Zou et al. (2025)	Representation-Based (logits nuclear norm + embedding diversity)	Subset-Relative (memory buffer)	Self Model	Submodular Subset (utility–diversity batch)	Online
ADAPT Zhao et al. (2026)	Representation-Based (similarity-based quality)	Anchor Comparison (quality/anchor similarity)	Proxy Model (online)	Soft Reweighting (adaptive LR)	Online
Diversity-as-a-Reward Ling et al. (2025)	Representation-Based (diversity reward; semantic entropy)	Subset-Relative (selected subset)	Self Model (probe)	Soft Reweighting (RL-like subset reward)	Online
AlpaGasus Chen et al. (2024)	Judge-Based (ChatGPT quality score)	Absolute Standard (quality threshold)	Judge Model (ChatGPT)	Hard Filtering (threshold filter, 52k→9k)	Offline
DEITA Liu et al. (2024b)	Judge-Based (Evol Complexity × Evol Quality + diversity)	Absolute Standard + Subset-Relative (abs. quality; dist. to selected)	Judge Model (ChatGPT-distilled scorers)	Submodular Subset (rank + Repr Filter)	Offline
MoDS Du et al. (2023)	Judge-Based (quality + coverage + necessity)	Subset-Relative (coverage + necessity)	Judge Model (reward model + SFT model)	Submodular Subset (quality→k-center→necessity)	Offline
CaR Ge et al. (2024)	Judge-Based (expert-aligned quality; 550M scorer)	Absolute Standard (quality + clustering)	Judge Model (550M scorer)	Submodular Subset (top-n + per-cluster)	Offline
SCAR Li et al. (2025)	Judge-Based (style consistency; form + surprisal)	Absolute Standard (within same-quality data)	Judge Model (style ranker)	Hard Filtering (most style-consistent)	Offline
#InsTag Lu et al. (2024)	Judge-Based (tag diversity + complexity; tag count)	Absolute Standard (tag distribution + count)	Judge Model (auto-tagging)	Submodular Subset (complexity-focused sampling)	Offline
Nuggets Li et al. (2024c)	Loss-Based (golden score; one-shot − zero-shot gain)	Self-Comparison (one- vs. zero-shot)	Self Model (ICL)	Hard Filtering (high golden-score)	Offline
SelectIT Liu et al. (2024a)	Loss-Based (multi-granularity self-uncertainty)	Self-Comparison (self-rated confidence)	Self Model (multi-LLM)	Hard Filtering (low-uncertainty)	Offline
Superfiltering Li et al. (2024a)	Loss-Based (IFD on a weak model)	Self-Comparison (w/ vs. w/o instruction, weak)	Proxy Model (weak, GPT-2)	Hard Filtering (top-k IFD)	Offline
S2L (SmallTo-Large) Yang et al. (2024)	Loss-Based (loss trajectory on a small model)	Subset-Relative (trajectory-similarity clusters)	Proxy Model (small)	Submodular Subset (cluster + even sampling)	Offline
LIMA Zhou et al. (2023)	Human judgment (quality + diversity; no automated metric)	Absolute Standard (human/expert standard)	Human Curation (human experts)	Manual (∼1000 hand-picked)	Offline (human)
DiverseEvol Wu et al. (2023)	Representation-Based (embedding distance to selected; diversity)	Subset-Relative (current subset, k-center)	Self Model (evolving embeddings)	Submodular Subset (k-center-greedy, multi-round)	Online
TAGCOS Zhang et al. (2024)	Gradient-Based (per-sample gradients as features)	Subset-Relative (full-set gradient coverage)	Self Model (gradients)	Submodular Subset (cluster + submodular coreset)	Offline
G-DIG Pan et al. (2024)	Gradient-Based (influence function + gradient diversity)	Anchor Comparison (high-quality seed)	Self Model (gradients + seed)	Submodular Subset (pos.-influence + cluster-resample)	Offline
SHED He et al. (2024)	Gradient-Based (Shapley value; marginal contribution)	Anchor Comparison (eval/task contribution)	Proxy Model (clusters + eval)	Hard Filtering (filter by Shapley)	Offline

continued on next page

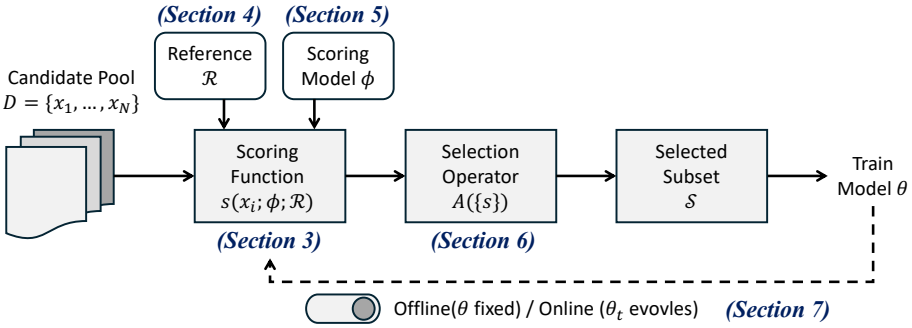

Figure 1 Pipeline of Data Selection for LLM Finetuning.

Table 1 (continued)

Method	① Metric	② Comparison	③ Model	④ Policy	⑤ Timing
DELIFT Agarwal et al. (2025)	Gradient-Based (ICL-inspired pairwise utility)	Subset-Relative (sample-to-sample)	Self Model (no gradient)	Submodular Subset (staged)	Offline
RDS+ Ivison et al. (2025)	Representation-Based (repr. similarity; pos.-weighted hidden states)	Anchor Comparison (target/task samples)	Proxy Model (frozen LM hidden states)	Hard Filtering (by repr. sim., scalable)	Offline
MIG Chen et al. (2025)	Representation-Based (information gain; label-graph space)	Subset-Relative (selected vs. full set)	Judge Model (external tags + embeddings)	Submodular Subset (greedy max info-gain)	Offline
QDIT Bukharin et al. (2023)	Representation-Based (quality + facility-location diversity)	Subset-Relative (subset representativeness)	Judge Model (ext. quality + embeddings)	Submodular Subset (greedy quality–diversity)	Offline
ZIP / Entropy Law Yin et al. (2024)	Representation-Based (compression ratio; lower = more diverse)	Subset-Relative (subset redundancy)	Model-agnostic (no scoring model)	Submodular Subset (greedy, lowest compression)	Offline

1.4 Organization

The rest of this survey is organized as follows. Section 2 gives a formal definition of the data selection problem and a unified notation. Sections 3–7 develop the five dimensions in turn, analyzing the design principles and representative works of each. Section 8 discusses open challenges, and Section 9 concludes and outlines future directions.

2 Problem Statement

2.1 Formalization

Let the candidate data pool be $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$, where each x_i is a training example—an (instruction, response) pair in the SFT setting, or an (instruction, chosen, rejected) triple in the preference-alignment setting. The goal of data selection is to choose, under a budget constraint, a subset $\mathcal{S} \subseteq \mathcal{D}$ such that the model M_θ (with parameters θ) trained on $\mathcal{S}$ maximizes expected performance on a target distribution or downstream task:

$$\mathcal{S}^\star \;=\; \arg \max_{\mathcal{S} \subseteq \mathcal{D},\, c(\mathcal{S}) \le B} \; \mathrm{Perf}\big(M_{\theta(\mathcal{S})}\big), \tag{1}$$

where $\theta(\mathcal{S})$ denotes the parameters obtained by training on $\mathcal{S}$, $c(\mathcal{S})$ is the cost of selection/training (e.g., number of examples, tokens, or FLOPs), B is the budget, and $\mathrm{Perf}(\cdot)$ measures performance on a validation/test distribution. In general, Eq. (1) is combinatorial and non-differentiable—model training is an expensive black box—so every practical method is some approximation of, or surrogate for, this objective.

2.2 Scoring Function and Selection Operator

Most methods decompose this intractable problem into two steps: first assign each sample a scalar value score, then apply a selection operator that chooses data based on scores.

(1) Scoring function $s(x_i; \phi, \mathcal{R})$**.** A scalar value for x_i, depending on the parameters ϕ of the *scoring model* (Dimension ③) and on a comparison/reference object $\mathcal{R}$ (Dimension ②). The concrete form of s defines Dimension ① (the metric).

Scoring model vs. trained model. A point that is easy to gloss over but central to this survey is that the scoring-model parameters ϕ are *not*, in general, the parameters θ of the model being trained. The two coincide only in the *self-scoring* regime, where the score is read off a (current or warmed-up) state of the target model itself, $\phi = \theta_t$. In the other regimes ϕ is a *different* model entirely: a separate—often smaller and cheaper—*proxy* ($\phi = \phi_p$), a strong *external* LLM or reward model acting as judge ($\phi = \phi_{\text{ext}}$), or, at the extreme, *no* parametric model at all (human curation). Conflating ϕ with θ would erase exactly the degree of freedom that Section 5 is about: *which model produces the score*, and hence how well the score aligns with the target model's training objective, how reusable it is, and what biases it inherits. We therefore keep ϕ and θ notationally distinct throughout.

(2) Selection operator $\mathcal{A}(\{s(x_i)\}) \to \mathcal{S}$**.** A map from scores to the final subset (or to sample weights). It may be a hard threshold/top-k rule, a continuous soft-weighting scheme, or an iterative greedy/search procedure; this is Dimension ④.

Dimension ⑤ (timing) then determines *when* these two steps occur, which is a separate axis from *which* model does the scoring: selection may run *before* training (offline, with ϕ held fixed) or be embedded *within* the training loop (online, where the scoring model tracks the evolving target model, $\phi = \theta_t$, so that s changes as training proceeds).

2.3 Offline vs. Online Selection

Offline selection is performed once before training: a fixed scoring model ϕ (a proxy, an external judge, or a frozen warmed-up snapshot θ_0 of the target model) scores the whole pool, a subset is chosen, and formal training proceeds. Its advantages are simplicity, a reusable subset, and decoupling from training; its drawback is that the scoring model's state differs from that of the final model—when $\phi \neq \theta$ this is the proxy/judge gap of Section 5, and even when $\phi = \theta_0$ the snapshot is a static one, so the chosen subset may not match the model's evolving state.

Online selection embeds scoring and selection in the training loop: at each (or every few) iteration(s), the current target-model state θ_t is itself the scoring model ($\phi = \theta_t$), scoring a candidate batch and dynamically selecting/reweighting it. This keeps scoring synchronized with the training state and aligned with the training objective, generalizing better, at the cost of per-step scoring overhead. The core engineering challenge of online methods is therefore to push scoring into the training loop while avoiding extra forward/backward passes.

2.4 Notation and Conventions

To unify the discussion we distinguish the two models explicitly. We write θ (or θ_t for its state at training step t) for the parameters of the *model being trained* (the target/final model of Eq. (1)), and ϕ for the *scoring-model* parameters, which take one of four regimes (Section 5): a proxy ϕ_p, the self ($\phi = \theta_t$), an external LLM/reward-model judge ϕ_{ext}, or none (human curation). For the comparison target we write $\mathcal{R}_{\text{self}}$ for a self-comparison reference (e.g., the with-/without-instruction versions of the same sample) and $\mathcal{R}_{\text{ext}}$ for an external anchor (validation set, anchor data, target distribution). A data selection method can thus be summarized as a tuple (metric, $\mathcal{R}$, ϕ, $\mathcal{A}$, timing). Using this notation, Sections 3–7 develop the five dimensions and map the surveyed works of Table 1 onto their respective values.

3 Dimension 1: Scoring Metric

The scoring metric is the heart of any data selection method, because it is what defines what counts as good data. Formally, it instantiates the scoring function $s(x_i; \phi, \mathcal{R})$ of Section 2, mapping each candidate to a scalar

that a downstream operator later turns into a selection. What truly separates one method from another is therefore not the operator but the *signal* that this scalar encodes. Four intuitions recur across the literature: some methods probe how hard a sample is for the model in its current state, some estimate a sample's causal contribution to a target objective, some ask how well a sample fits or broadens a distribution, and some defer to an external notion of quality. Together with a fifth intuition that is specific to preference alignment, these organize the surveyed works into the five schools below. We treat each school in turn, emphasizing the mechanism that gives it its distinctive view of data value rather than cataloguing methods one by one, and we return in Sections 3.6 and 3.7 to two patterns that cut across all five. Table 2 summarizes the schools.

Five Scoring-Metric Schools

1. **Loss-Based** (Section 3.1): a sample is valuable to the extent that the model can *currently* learn from it, read off as a conditional loss, a loss *gap*, or an uncertainty estimate.
2. **Gradient-Based** (Section 3.2): a sample's value is its *marginal contribution* to target or validation performance, measured via gradient inner products, influence functions, optimal transport, or Shapley values.
3. **Representation-Based** (Section 3.3): a sample is scored by how well it *fits* the model's distribution, how *similar* its representation is to a target, or how much *coverage* it adds.
4. **Judge-Based** (Section 3.4): quality comes from a strong *external* judge, either an LLM-as-judge or a purpose-trained scorer, rather than from the trained model's own loss or gradients.
5. **Preference Margin** (Section 3.5): specific to alignment, a preference pair is scored by its reward or DPO *margin*, on the premise that confidently ordered pairs are the most informative and the least noisy.

3.1 Loss-Based Signals

The premise of this school is that a sample is valuable to the extent that "the model can currently learn from it." All such signals are, at heart, a conditional loss, a loss *gap*, or an uncertainty estimate read off the model.

Loss gaps from an auxiliary cue. The cleanest instance is IFD / Cherry-LLM Li et al. (2024b), whose *Instruction-Following Difficulty* score $\mathrm{IFD}(x) = \frac{\mathcal{L}(r|q)}{\mathcal{L}(r)}$ is the ratio of the conditional loss of generating the response r given the instruction q to the unconditional loss of generating r alone. A high ratio means the instruction helps little, since the model already produces r on its own, whereas a very low ratio flags a trivial sample, so IFD favors the mid-difficulty examples whose loss the instruction genuinely reduces. MIWV Jiang et al. (2025) applies the same logic to a different cue. Its *Model Instruction Weakness Value* is the drop in response loss when a one-shot in-context example is added to a zero-shot prompt, and a large drop marks an instruction on which the model is weak and therefore benefits most from a demonstration, with the authors reporting that the top one percent by MIWV already beats training on the full set. Nuggets Li et al. (2024c) carries the gap idea from a single cue to a whole probe set, defining a sample's *golden score* as the average zero-shot to one-shot performance gain it produces when used as the in-context exemplar, and again keeps only the highest-scoring examples.

Difficulty and intrinsic uncertainty. A second group reads the loss not as a gap but as a direct measure of how hard or how uncertain a sample is. Selective DPO Gao et al. (2025) works in the preference setting, defining a pair's *difficulty* as the reference model's validation loss on it, equivalently its position in the learning order that stays consistent across runs, and then filters out the overly hard pairs whose noise it shows degrades alignment, letting the model's capacity set the difficulty threshold. LEAD Lin et al. (2025) turns this static reading into a dynamic one through its *Instance-level Dynamic Uncertainty*, which fuses the instantaneous training loss, a first-order gradient-approximated loss change, and an exponential moving average of historical loss so that utility is estimated inside the standard training loop with no extra inference. SelectIT Liu et al. (2024a) removes the external reference entirely and instead aggregates the model's own intrinsic uncertainty at three granularities, namely token-probability spread, sentence-level self-rating, and cross-model self-reflection.

Cheap offline proxies. Because a forward pass is far cheaper than a gradient, a third group pushes the same loss-based intuition toward near-zero cost. Superfiltering Li et al. (2024a) observes that the relative IFD

ranking is remarkably stable across model scales, so an IFD computed by a tiny GPT-2-scale model transfers weak-to-strong and filters data almost for free. S2L Yang et al. (2024) instead records each sample's *loss trajectory* over training on a small reference model and clusters the trajectories, sampling evenly from the clusters so that selection is driven by the loss dynamics rather than by a single snapshot. Both buy a large reduction in cost at the price of some fidelity to the final model, a trade we return to in Section 3.7.

3.2 Gradient-Based Signals

This school characterizes a sample's *contribution* to target/validation performance through gradients (inner products, influence functions, optimal transport, Shapley values). It is the most theoretically grounded but typically the most expensive, since it requires per-sample gradients or repeated retraining.

Gradient alignment with a target. The canonical method is LESS Xia et al. (2024), which estimates a candidate's first-order *influence* as the inner product between its gradient, Adam-corrected and randomly projected to a low rank, and the averaged gradient of a few target-task validation examples, then takes the top-k. The selection it produces is thus explicitly targeted at a downstream task. GREATS Wang et al. (2024) preserves this inner-product idea but makes it affordable at every iteration, approximating through a first-order Taylor expansion how much each point in a batch would reduce held-out validation loss and using a ghost inner-product trick to read all per-sample scores off quantities already present in the backward pass. CLUES Zhao et al. (2024) accumulates the same signal over time, scoring a sample by the trace across training of the inner product between its gradient and that of a high-quality anchor set, on the premise that valuable data follows training dynamics similar to the anchor, which makes it a natural fit for collaborative and private-domain settings.

From alignment to distribution and budget. Two methods reframe what the gradient is aligned against. GOT-D Kang et al. (2024) adopts an optimal-transport view, scoring each candidate by how much its gradient shifts the pretraining distribution toward the target distribution and thereby selecting the data that most efficiently warms up fine-tuning. CADS Wan et al. (2025) instead folds the compute budget into the objective itself, casting selection as a bilevel optimization whose inner loop trains within a fixed budget and whose outer loop optimizes the selection by a Hessian-free policy gradient, so that the importance signal is tied directly to what the budget allows.

Coreset and game-theoretic extensions. A final group reuses gradients offline as the raw material for building a subset. TAGCOS Zhang et al. (2024) treats per-sample gradients as feature representations, clusters them, and greedily maximizes a submodular coreset objective so that the chosen subset covers the full-data gradient. G-DIG Pan et al. (2024) pairs an influence-function filter, which keeps samples with beneficial influence on a small high-quality seed set, with gradient-cluster resampling for diversity. SHED He et al. (2024) estimates each sample's *Shapley value*, its average marginal contribution to performance, through proxy clusters and Monte-Carlo evaluation, and the resulting selections transfer across models. DELIFT Agarwal et al. (2025) keeps the influence intuition but drops the gradients, defining an ICL-inspired pairwise utility kernel that measures how much one example, used as context, lowers the loss on another, and optimizing a submodular objective over it, which makes the whole idea far cheaper.

3.3 Representation-Based Signals

This school scores how well a sample *fits* the model's distribution, how *similar* its representation is to a target, or how much *diversity/coverage* it adds to the selected subset.

Fit to a model or a task. GRAPE Zhang et al. (2025a) gathers, for each instruction, candidate responses from several LLMs and keeps the one with the highest likelihood under the target base model, so that the training data matches the model's own pretrained distribution rather than being distilled from a possibly mismatched teacher. RDS+ Ivison et al. (2025) measures fit against a task instead of the model, scoring representation similarity to target samples through a position-weighted mean pooling of a frozen LM's hidden states. This off-the-shelf embedding is cheap enough to scale to millions of candidates and has been shown to beat many heavier methods at that scale.

Coverage in embedding space. A second group scores a sample by how much it broadens the selected set rather than how well it fits a fixed reference. Diversity-as-a-Reward Ling et al. (2025) measures diversity through the semantic entropy of a trainable probe and uses it as an RL-style reward. UDS Zou et al. (2025) combines two terms, a utility score given by the nuclear norm of the logits matrix, which captures confidence together with intra-sample diversity, and an inter-sample diversity score given by embedding distance to a lightweight memory buffer. DiverseEvol Wu et al. (2023) grows the set by k-center distance in the model's evolving embedding space, so that each new sample is the one farthest from those already chosen. Sitting between fit and coverage, Priority-on-High-Quality Zhang et al. (2025b) estimates quality from response consistency under noise injection, on the view that a stable response signals a robust sample, and then combines this quality score with inter- and intra-class diversity.

3.4 Judge-Based Signals

A large family of SFT methods obtains a *quality* signal not from the trained model's own loss or gradients but from a strong external judge, either an *LLM-as-judge* or a purpose-trained scorer. The appeal is that a capable judge encodes a notion of quality that is hard to express as a loss or a gradient, and the price, examined in Section 5, is that the signal inherits whatever the judge values.

Prompting a general-purpose judge. The simplest approach prompts a capable general model to rate the data directly. AlpaGasus Chen et al. (2024) asks ChatGPT to assign each instruction, input, and response triple an integer quality score and keeps only the high-scoring ones, pruning Alpaca from 52k to 9k while actually *improving* the tuned model. #InsTag Lu et al. (2024) puts the same judge to a different use, prompting ChatGPT to auto-tag instructions with fine-grained intent and semantic tags and then scoring complexity by the tag count and diversity by the tag distribution.

Purpose-built scorers. A second group replaces the general judge with a scorer trained for the job, trading some generality for control and a lower inference cost. DEITA Liu et al. (2024b) distills ChatGPT ratings over Evol-Instruct variants into dedicated scorers for *Evol Complexity* and *Evol Quality* and combines their product with an embedding-distance diversity filter. MoDS Du et al. (2023) uses a reward model for quality and then layers two further criteria on top, namely type coverage through k-center selection and necessity, where a sample counts as necessary if an initial SFT model still answers it poorly. CaR Ge et al. (2024) ranks with a small 550M expert-aligned scoring model and preserves diversity by selecting representatives per cluster. SCAR Li et al. (2025) ranks responses by *style consistency*, combining linguistic form with instructional surprisal, on the hypothesis that stylistically coherent training data tunes more efficiently. Across all of these, the signal is strong and human-aligned, but it inherits the judge's biases and its API dependence, a point we take up in Section 5.

3.5 Preference-Margin Signals (Alignment-Specific)

Alignment introduces a signal that has no analogue in SFT, namely how cleanly a preference pair is ordered. Less-is-More Deng et al. (2025) scores a pair by its *reward margin*, fusing an external reward model's margin with the implicit DPO margin through a Bayesian aggregation, and then keeps the large-margin pairs on the principle that confidently ordered pairs are the most informative and the least noisy. This view is the mirror image of Selective DPO's difficulty view (Section 3.1), since margin and difficulty are two ways of asking the same question of which preference pairs are worth training on.

3.6 Quality $\times$ Diversity as a Joint Objective

A striking pattern across the SFT methods is that quality is rarely used on its own. The reason is structural rather than incidental. A pure quality ranking tends to concentrate on a narrow region of high-scoring but near-duplicate examples, so almost every method couples its quality score with an explicit diversity or coverage term that rewards spreading the budget across the data. DEITA's Repr Filter, MoDS's k-center coverage, CaR's per-cluster selection, #InsTag's tag-distribution diversity, S2L's trajectory-cluster sampling, and DiverseEvol's k-center growth are all instances of this coupling. A few methods go further and fold the two goals into one objective. QDIT Bukharin et al. (2023) maximizes a linear combination of a quality score

Table 2 Comparison of the five scoring-metric schools. "Cost" is the order-of-magnitude compute to score the pool; "Target alignment" is how directly the metric approximates the training objective.

School	Core signal	Cost	Target alignment	Representative works
Loss-based	Conditional loss, loss gap, or uncertainty	Low–med.	Indirect (proxy for learnability)	IFD Li et al. (2024b), MIWV Jiang et al. (2025), Selective DPO Gao et al. (2025), LEAD Lin et al. (2025), SelectIT Liu et al. (2024a), Superfiltering Li et al. (2024a), S2L Yang et al. (2024)
Gradient-based	Marginal contribution to target/val. performance	High	Direct	LESS Xia et al. (2024), GREATS Wang et al. (2024), CLUES Zhao et al. (2024), GOT-D Kang et al. (2024), CADS Wan et al. (2025), TAGCOS Zhang et al. (2024), G-DIG Pan et al. (2024), SHED He et al. (2024), DELIFT Agarwal et al. (2025)
Representation-based	Fit to model distribution, embedding similarity, coverage	Low–med.	Indirect (fit/coverage)	GRAPE Zhang et al. (2025a), RDS+ Ivison et al. (2025), UDS Zou et al. (2025), DiverseEvol Wu et al. (2023), Div.-as-Reward Ling et al. (2025), Priority-HQ Zhang et al. (2025b)
Judge-based	Human-aligned quality from a strong scorer	Med. (API)	Human-aligned, not model-specific	AlpaGasus Chen et al. (2024), DEITA Liu et al. (2024b), MoDS Du et al. (2023), CaR Ge et al. (2024), SCAR Li et al. (2025), #InsTag Lu et al. (2024)
Preference margin	Reward / DPO margin of a pair	Med.	Alignment-specific	Less-is-More Deng et al. (2025)

and a *facility-location* term that rewards how well the subset represents the full pool. MIG Chen et al. (2025) maximizes *information gain* over a label-graph semantic space, so that a sample covering an under-represented region contributes more than a redundant one. ZIP Yin et al. (2024), drawing on an entropy-law argument, treats the subset's *compression ratio* as a model-agnostic proxy for redundancy and greedily assembles a set that compresses poorly and is therefore diverse. The recurring lesson is that a single scalar quality metric is seldom enough by itself.

3.7 The Compute-Cost Spectrum of Metrics

Metrics also differ sharply in cost, and the cost tracks how directly a metric approximates the true quantity of interest, namely how useful a given datum is to the model. The gradient, influence, and Shapley signals sit at the expensive end (LESS, TAGCOS, G-DIG, SHED, and the online GREATS). They are closest to the training objective but require per-sample gradients or repeated evaluation. The loss and uncertainty signals sit in the middle (IFD, MIWV, SelectIT, LEAD), needing only one or two forward passes per sample. The ICL, representation, and weak-model proxies sit at the cheap end (Nuggets, Superfiltering, RDS+, S2L, ZIP). Superfiltering's weak-model IFD and RDS+'s frozen-LM embeddings avoid training-time gradients altogether, and ZIP needs no model at all. This spectrum is the metric-level counterpart of the compute-budget argument in Section 7, and a recurring empirical finding, reported for example by RDS+ and by Compute-Constrained Data Selection, is that the cheap end is often competitive once the compute spent on selection is itself counted.

3.8 Summary

The five schools reflect different philosophies of what data value means. The loss and difficulty view cares about the model's current learning state and is cheap but indirect. The gradient, influence, and Shapley view approximates marginal contribution directly, at high cost. The distribution and representation view emphasizes fit and coverage. The external-judge family imports human-aligned quality from a strong model and pays for it by inheriting that model's biases. The preference-margin school serves alignment in particular.

Two trends cut across all of them (Table 2). Quality is almost always coupled with diversity, and the cheaper proxy metrics increasingly chase, and often match, the same target that the expensive gradient-based ones pursue.

4 Dimension 2: Comparison Target

No score is meaningful in isolation. Every score is computed against some baseline $\mathcal{R}$ in the scoring function $s(x_i; \phi, \mathcal{R})$, and the choice of that baseline is as consequential as the choice of metric. It decides three things at once: whether a method needs extra annotation or reference data, whether its selection can be steered toward a particular downstream task, and whether diversity can be expressed as a selection criterion at all. Sorting methods by the nature of this baseline yields the four kinds of comparison below, which range from a contrast built entirely inside the sample to a contrast against the subset already chosen.

Four Comparison Targets

1. **Self-Comparison** (Section 4.1): the contrast is constructed purely *within* the sample or model, for example loss with versus without the instruction, so no external validation or reference data is required.
2. **Anchor Comparison** (Section 4.2): the reference is an *external task anchor* such as a validation set, a high-quality seed, or a target distribution, which enables *targeted*, task-specialized selection.
3. **Absolute Standard** (Section 4.3): the sample is judged against a *fixed quality scale* such as an LLM judge or a human standard, asking whether it is a good example rather than whether it is good *for my task*.
4. **Subset-Relative** (Section 4.3): the reference is the *already-selected subset*, so a sample's marginal value depends on what has been chosen, which is what turns diversity into a first-class selection criterion.

4.1 Self-Comparison: No External Data Required

Self-comparison methods construct a contrast purely within the sample or the model, and therefore need no external validation set or reference data, giving them a low deployment barrier and broad applicability.

The most common internal contrast removes a cue from the input and measures the resulting change in loss. IFD Li et al. (2024b) takes the reference to be the same response without the instruction, reading the conditional loss $\mathcal{L}(r \mid q)$ against the unconditional loss $\mathcal{L}(r)$. MIWV Jiang et al. (2025) removes a different cue, comparing the one-shot response against the same instruction stripped of its ICL exemplar, and Nuggets Li et al. (2024c) applies the same zero-shot to one-shot contrast on a probe set, with every quantity produced by the model itself. A second kind of contrast looks inward across time or confidence rather than across an edited input. LEAD Lin et al. (2025) uses the sample's own loss-dynamics history, its EMA and gradient-approximated change over training, as the reference, whereas SelectIT Liu et al. (2024a) relies on the model's own confidence and self-rating with no external judge at all. A third kind makes the candidates compete with one another. In GRAPE Zhang et al. (2025a) the candidate responses to a single instruction are ranked against each other under the base model's likelihood, an internal contest that needs no external scale.

4.2 Anchor Comparison: Dependence on External References

Anchor-comparison methods introduce an external reference, typically a target-task validation set, high-quality anchor data, or a target distribution. The stronger and the closer this anchor is to the downstream task, the more the method can perform *targeted* selection, yielding data sharply specialized to that task.

What distinguishes these methods is how the external reference is expressed, and the choices form a progression from concrete data toward an abstract scale. The most direct form is a validation-set gradient. LESS and GREATS Xia et al. (2024); Wang et al. (2024) treat the gradient of the target-task validation set as the anchor and score a sample by how well its own gradient aligns with reducing validation loss, which is exactly what makes both methods explicitly task-targeted. A looser form replaces the validation gradient with a high-quality reference set and matches a sample's behavior to it. CLUES Zhao et al. (2024) uses a server-aggregated high-quality reference set and scores the similarity of *training dynamics* to that anchor, and G-DIG Pan et al.

Table 3 The four comparison targets and their trade-offs.

Reference	External data?	Selection style	Representative works
Self	None	Generic quality, broadly applicable	IFD Li et al. (2024b), MIWV Jiang et al. (2025), Nuggets Li et al. (2024c), GRAPE Zhang et al. (2025a), SelectIT Liu et al. (2024a), LEAD Lin et al. (2025)
Task anchor	Validation / seed / target dist.	Targeted, task-specialized	LESS Xia et al. (2024), GREATS Wang et al. (2024), CLUES Zhao et al. (2024), G-DIG Pan et al. (2024), GOT-D Kang et al. (2024), RDS+ Ivison et al. (2025), Instruction Mining Cao et al. (2023)
Absolute standard	A judge or human scale	Generic, human-aligned quality	AlpaGasus Chen et al. (2024), DEITA Liu et al. (2024b), CaR Ge et al. (2024), SCAR Li et al. (2025), #InsTag Lu et al. (2024), LIMA Zhou et al. (2023)
Selected subset	None (internal)	Marginal / coverage, anti-redundancy	MoDS Du et al. (2023), DEITA Liu et al. (2024b), QDIT Bukharin et al. (2023), MIG Chen et al. (2025), UDS Zou et al. (2025), DiverseEvol Wu et al. (2023)

(2024) uses a small high-quality seed set as the positive-influence reference. Other methods raise the anchor from a sample set to a whole distribution. GOT-D Kang et al. (2024) takes the target distribution itself as the anchor and measures how much a sample shifts the pretraining distribution toward it, whereas RDS+ Ivison et al. (2025) compares each candidate's representation to target or task samples. At the most abstract end, the anchor is an external scoring model rather than data, as in Less-is-More and Instruction Mining Deng et al. (2025); Cao et al. (2023), which read each sample's margin or predicted loss against the scale of an external reward or quality estimator.

4.3 Absolute Standards and Subset-Relative Comparison

Two further references fit neither pure self-comparison nor a task-specific anchor. The first is an *absolute quality standard*, used by the external-judge methods of Section 3.4. AlpaGasus, DEITA, CaR, SCAR, and #InsTag compare each sample to a fixed quality scale, and LIMA compares it to an expert human standard, rather than to the model or to any task set. Such a standard is task-agnostic, since it asks whether a sample is a good example in general rather than whether it is good for one particular task. This is exactly why these methods generalize broadly but cannot specialize.

The second is the *already-selected subset* itself, used by the diversity-aware methods. MoDS's k-center coverage, DEITA's Repr Filter, QDIT's facility-location representativeness, MIG's information gain over the selected set, UDS's memory buffer, and DiverseEvol's k-center-greedy distance all measure a sample's *marginal* value relative to what has already been chosen. This subset-relative reference is what promotes diversity from a per-sample score to a first-class selection criterion. Because a sample's worth now depends on its neighbors in the selection, the reference is dynamic and the operator can no longer be a one-shot ranking but must proceed sequentially (cf. Section 6).

4.4 Trade-off: Generality vs. Specificity

Underneath the four references runs a single tension between generality and specificity, summarized in Table 3. Self-comparison and absolute-standard methods need no target-task data and apply broadly, but for the same reason they can only perform generic quality filtering and struggle to optimize for a specific downstream task. Anchor-comparison methods sit at the opposite pole. They enable precise, targeted selection, yet their effectiveness hinges on the quality and representativeness of the anchor and they require extra annotated reference data. The trade-off is monotone: the stronger the anchor, the more targeted the selection, but also the heavier the dependence on annotation and the greater the risk of amplifying the anchor's own bias (Section 8). Subset-relative comparison is orthogonal to this axis and is almost always layered on top of a per-sample quality score to control redundancy rather than used alone. In practice the choice comes down to whether a clear target task exists and whether high-quality reference data can be obtained for it.

5 Dimension 3: Scoring Model

Every score has to be produced by some model, and which model does the producing is a design axis in its own right rather than an implementation detail. At first glance the choice looks binary, a separate *proxy* model or the *self*, meaning the model being trained. Across the surveyed works, however, it is better seen as a spectrum of four regimes, running from a deliberately weak proxy at one end to expensive human curation at the other. Where a method sits on this spectrum governs four practical properties at once: how consistent the score is with the training objective, how much the scoring costs, how reusable the scores are across runs, and what kind of bias they carry.

Four Scoring-Model Regimes

1. **Proxy Model** (Section 5.1): scores with a separate, usually smaller and sometimes deliberately *weak* model. Offline and reusable, but exposed to a *proxy–target gap*.
2. **Self Model** (Section 5.2): scores with the model being trained, so scoring is perfectly aligned with the training objective and naturally online, at a per-step compute cost.
3. **Judge Model** (Section 5.3): scores with a strong external model or a purpose-trained scorer. Human-aligned and reusable, but API-dependent and prone to importing the judge's bias.
4. **Human Curation** (Section 5.4): no scoring model at all, as experts hand-pick examples. The highest-fidelity signal, but with no scalability.

5.1 Proxy Models

The proxy route scores data with a model that is deliberately decoupled from the one being trained, and its appeal is structural. Because the proxy never shares parameters with the target model, scoring can run entirely before training, its scores can be cached and reused across many runs, and not a single forward or backward pass is charged to the training budget.

Independent proxies. The most common choice is simply a separate, smaller model. LESS scores with a small LoRA model whose gradients stand in for the target's Xia et al. (2024), Instruction Mining reads quality off an external reward model Cao et al. (2023), CADS trains a budget-constrained inner model whose behavior approximates training on a candidate subset Wan et al. (2025), and GOT-D uses a distribution-matching model to measure how a sample moves the data distribution toward the target Kang et al. (2024). In each case the proxy is chosen to be cheaper than the target while still carrying the particular signal the method relies on.

Deliberately weak proxies. A bolder line of work pushes the proxy far below the target in scale and argues that this costs almost nothing. The enabling observation, made sharply by Superfiltering Li et al. (2024a), is that selection needs the relative ranking of samples rather than their absolute scores, and this ranking is remarkably stable across model scales. An IFD score computed by a GPT-2-scale model therefore transfers weak-to-strong and filters data for a tiny fraction of the cost. S2L applies the same reasoning to loss trajectories recorded on a small reference model Yang et al. (2024), and SHED approximates Shapley values through proxy clusters rather than the full model He et al. (2024). The common bet is that the signal the proxy carries is largely scale-invariant, so cheapness need not cost accuracy.

Pros and Cons. The advantages all follow from decoupling. Scoring is offline, the resulting scores and gradient stores are reusable, and a weak proxy can make the whole step nearly free. The liability also follows from decoupling and is the defining weakness of the regime. The *proxy–target gap* arises because the proxy differs from the final model in scale or training state, so its notion of good data need not match the target's. This gap tends to be small for ranking-stable quality signals such as IFD, where Superfiltering's transfer holds up, but it grows for signals that depend on the exact parameter state, such as gradient and influence estimates, and it widens monotonically as proxy and target diverge in scale.

5.2 Self Models

The self route scores data with the model being trained, which makes the score and the training objective one and the same. It is the most populous regime, and within it two sub-modes are worth separating because they behave very differently in practice.

Warm-up scoring versus in-loop scoring. In the first sub-mode the model scores data from a fixed, lightly warmed-up snapshot, after which training proceeds on the chosen subset. IFD reads its difficulty score off a briefly pre-experienced model Li et al. (2024b), GRAPE ranks responses under the base model's likelihood Zhang et al. (2025a), and Nuggets scores with the model in a one-shot probe Li et al. (2024c). Here the self model is used much like a proxy that happens to be a snapshot of the target, and selection remains offline. In the second sub-mode the score is read from the live parameters θ_t as training runs, so the data chosen at step t reflects the model's current state. GREATS, LEAD, and UDS all operate this way and are therefore naturally online Wang et al. (2024); Lin et al. (2025); Zou et al. (2025). The remaining self-scoring methods, among them SelectIT, TAGCOS, DELIFT, and DiverseEvol, fall between these poles, scoring with the model but not necessarily at every step Liu et al. (2024a); Zhang et al. (2024); Agarwal et al. (2025); Wu et al. (2023).

The model as its own probe. Diversity-as-a-Reward Ling et al. (2025) takes the idea to its logical end by casting the LLM in a dual role, as the output probe that scores and selects data and as the input that is then fine-tuned on the result, so the same network both judges and is judged.

Pros and Cons. The decisive advantage is perfect alignment, since the score is produced by exactly the model whose performance the selection is meant to improve, and because it tracks the live state it supports online, training-adaptive selection. The matching cost is that every scoring step spends the target model's own compute, which is why LEAD, GREATS, and UDS work so hard to fold scoring into the standard loop without extra inference or backprop, through devices such as the ghost inner-product, logits reuse, and the in-loop IDU estimate. A subtler risk is a feedback loop. Because the model selects the very data that will shape its next state, any bias already present in the model can be reinforced rather than corrected, so self-scoring methods that lack an external anchor usually lean on a diversity term to stay honest.

5.3 Judge Models

A third regime, prominent in instruction tuning, hands scoring to a strong *external* model that is neither a cheap proxy of the trained model nor the trained model itself. What unites its members is that the authority on quality lives outside the proxy-versus-self axis entirely, in a model chosen precisely because it is more capable or better aligned than the one being trained.

General-purpose LLM judges. The most direct form prompts a powerful general model to rate data on the fly. AlpaGasus asks ChatGPT for an integer quality score per example Chen et al. (2024), and #InsTag prompts the same model to tag instructions with fine-grained intent labels from which complexity and diversity are then read Lu et al. (2024). The signal is immediate and needs no training of a scorer, but every score is an API call.

Distilled and purpose-trained scorers. A second form removes the recurring API cost by baking the judge into a local model. DEITA distills ChatGPT ratings into dedicated Evol-Complexity and Evol-Quality scorers Liu et al. (2024b), MoDS scores with a reward model Du et al. (2023), and CaR and SCAR train compact ranking models for quality and style respectively Ge et al. (2024); Li et al. (2025). Distillation converts a slow, costly, and drifting external judge into a fast, reusable, and reproducible one, at the price of freezing whatever the teacher happened to believe about quality.

Pros and Cons. The quality signal is strong and well aligned with human preference, often issued by a model far more capable than the target, and once distilled it is cheap and reusable. The costs are equally characteristic. A general judge depends on an external API whose behavior can shift between runs and so

Table 4 The four scoring-model regimes. The regime governs reusability, alignment with the training objective, the kind of bias inherited, and the natural timing.

Regime	Cost / reuse	Main weakness	Representative works	Timing
Proxy (incl. weak)	Cheap, reusable, no train compute	Proxy–target gap	LESS Xia et al. (2024), GOT-D Kang et al. (2024), CADS Wan et al. (2025), Superfiltering Li et al. (2024a), S2L Yang et al. (2024), SHED He et al. (2024)	Offline
Self (trained model)	Per-step train compute	Consumes training budget	IFD Li et al. (2024b), GRAPE Zhang et al. (2025a), MIWV Jiang et al. (2025), GREATS Wang et al. (2024), LEAD Lin et al. (2025), UDS Zou et al. (2025), Nuggets Li et al. (2024c), TAGCOS Zhang et al. (2024), DELIFT Agarwal et al. (2025)	Online/offline
Judge model	Med. (API/scorer), reusable	Judge bias, API dependence	AlpaGasus Chen et al. (2024), DEITA Liu et al. (2024b), MoDS Du et al. (2023), CaR Ge et al. (2024), SCAR Li et al. (2025), #InsTag Lu et al. (2024)	Offline
Human curation	Very costly, not scalable	No scalability/reproducibility	LIMA Zhou et al. (2023)	Offline

undermines reproducibility, and every judge imports its own biases, with a well-documented tendency to reward verbosity, surface style, and particular formats. Most fundamentally, the scores reflect the judge's notion of quality, which is not guaranteed to coincide with what most improves the target model.

5.4 Human Curation (No Scoring Model)

At the far end of the spectrum there is no scoring model at all. LIMA selects roughly a thousand prompts and responses entirely by hand, curated by experts for quality, difficulty, and diversity Zhou et al. (2023). Its result motivates much of this survey, since a small, meticulously curated set can match or surpass tuning on far larger noisy corpora, which the authors read as evidence for a *superficial alignment hypothesis*, the claim that fine-tuning mainly teaches format and style while knowledge is already acquired in pretraining. Human curation thus supplies the highest-fidelity signal available and a natural upper bound against which automatic scorers can be measured. Its limitation is equally clear. Manual curation does not scale, is expensive to reproduce, and cannot be retargeted to a new task without repeating the effort, which is exactly the pressure that drives the rest of this dimension toward automatic proxies, judges, and self-scoring.

5.5 Trade-off: Reusability, Consistency, Alignment, and Cost

The four regimes trade off the same four properties differently (Table 4). Proxy and external-judge scoring are offline, reusable, and free of training-compute cost, but they introduce a proxy–target gap and a judge-bias gap respectively. Self scoring is perfectly consistent with the training objective and naturally online, at a per-step compute cost. Human curation gives the highest-fidelity signal of all and no scalability whatsoever. The practical consequence is a clean division of labor. External judges dominate when human-aligned quality matters and an API is available, weak proxies dominate when the goal is cheap, reusable, large-scale filtering, and self scoring dominates when selection must stay synchronized with the training state. The dimension is therefore tightly coupled with timing (Section 7), in that proxy, judge, and human scoring lean offline while self scoring leans online. Underneath the four regimes lies a single trade between fidelity and scalability. Human curation maximizes fidelity and gives up scale, weak proxies do the reverse, and judge and self scoring occupy the middle from opposite directions. Nothing forbids combining them, and several methods do, using an external judge for quality while leaning on the target model for task targeting, so the regimes are better read as ingredients than as mutually exclusive choices.

6 Dimension 4: Selection Policy

Once sample scores are in hand, they still have to be turned into actual training data, and this is the job of the *selection operator*. The operator converts a scalar signal into a concrete subset, and its form decides something the score alone cannot, namely whether interactions among samples such as redundancy, complementarity, and coverage are captured or ignored. A per-sample cut treats every example as if it stood alone, whereas a subset-aware operator can reason about the set as a whole. Ordered by exactly this capacity, the operators fall into the four families below, from a one-shot top-k cut to sequential, subset-aware optimization.

Four Selection-Policy Families

1. **Hard Filtering** (Section 6.1): sort by score and take the top-k or apply a threshold. Simple and controllable, but per-sample and diversity-blind.
2. **Soft Reweighting** (Section 6.2): discard nothing and instead assign each sample a continuous weight or learning rate, letting low-value samples contribute smaller gradients.
3. **Optimization** (Section 6.3): treat selection as a sequential or combinatorial optimization, capturing within-batch interactions and jointly tuning the subset size.
4. **Submodular Subset** (Section 6.4): greedily assemble a representative, low-redundancy subset by k-center, submodular, or information-gain objectives, making diversity the explicit goal.

6.1 Hard Filtering: Top-k / Threshold

Hard filtering is the most direct and by far the most common operator. In its purest form one sorts every sample by its score and keeps a fixed number of the best, the top-k rule, or sets a score threshold and discards everything below it. A small but important generalization keeps an interval rather than a tail. IFD retains mid-range scores because both the easiest and the hardest examples are unhelpful Li et al. (2024b), and Selective DPO removes an overly hard tail rather than a low-quality one Gao et al. (2025). The remaining offline methods are mostly plain top-k or threshold rules, among them LESS by influence Xia et al. (2024), MIWV by instruction weakness Jiang et al. (2025), and GRAPE by per-instruction likelihood Zhang et al. (2025a). The choice between a fixed k and a floating threshold is itself meaningful, because a fixed k pins the budget exactly but ignores how quality varies across pools, whereas a threshold lets the selected size grow or shrink with the data at the cost of one more parameter to calibrate.

Pros and Cons. The strategy is trivial to implement, transparent, and gives precise control over the budget. Its defining weakness is that it scores and keeps each sample in isolation, so it is blind to the relationships among the samples it keeps. Two near-identical high-scoring examples are both retained while a useful but lower-scoring outlier is dropped, which is why hard filtering on its own tends to produce high-quality yet redundant subsets and is so often paired with a separate diversity stage. The threshold or value of k also usually has to be tuned, and a threshold calibrated on one distribution rarely transfers unchanged to another.

6.2 Soft Reweighting: Discard Nothing

Soft reweighting refuses the discrete choice altogether. Instead of deciding whether each sample is in or out, it gives every sample a continuous weight, so that a low-value sample contributes a smaller gradient rather than disappearing. Formally this relaxes the binary subset indicator into a continuous weight, which links selection to importance weighting and to curriculum learning, where the weights may also evolve as training proceeds. ADAPT realizes this by turning a similarity-based quality signal into a per-sample adaptive learning rate, an online reweighting that never changes the sample count Zhao et al. (2026), and Diversity-as-a-Reward weights each sample by its diversity reward Ling et al. (2025). Because the weights are most naturally recomputed as the model changes, soft reweighting sits comfortably inside the training loop and is closely tied to online timing (Section 7).

Pros and Cons. Keeping all data avoids the hard discarding of information and preserves the diversity of the pool, and a smooth weight can express shades of value that a binary cut cannot. The costs are that a

score must still be computed for every sample, since nothing is filtered away in advance, that the weighting scheme is harder to design than a threshold, and that poorly calibrated weights can destabilize training when a few samples come to dominate the gradient.

6.3 Optimization

A third family stops treating scores as a ranking to be cut and instead frames selection as an explicit optimization problem, which lets the operator reason about a whole batch or budget at once. The methods differ mainly in what they optimize and how. GREATS performs greedy batch selection at every iteration, adding to the batch the point whose estimated marginal effect on validation loss is largest Wang et al. (2024). LEAD runs a two-stage coarse-to-fine search in which a multi-armed bandit first decides which data clusters are worth sampling and an instance-level uncertainty score then refines the choice within them, which amortizes the cost of scoring across many samples Lin et al. (2025). Instruction Mining treats the size of the subset as a hyperparameter and tunes it by Bayesian search with BlendSearch Cao et al. (2023), and CADS casts the whole problem as a bilevel optimization whose outer level selects data while the inner level trains under a budget Wan et al. (2025). What these share is that the selected set is the solution to a search rather than the output of a sort.

Pros and Cons. Because they optimize over sets or budgets, these operators capture interactions that per-sample rules miss, such as redundancy and complementarity within a batch, and they can tune the subset size jointly with the selection, so in principle they reach better solutions. The price is real. The search is complex to implement, often expensive, and usually tractable only through first-order approximations or greedy heuristics that trade exactness for speed.

6.4 Submodular Subset Selection

A fourth family, dominant among the diversity-aware SFT methods, redefines the goal itself. Rather than scoring samples independently and hoping the result is varied, it directly seeks a representative, low-redundancy subset, which makes diversity a property of the chosen set instead of an afterthought. All of these operators are inherently subset-relative, since a sample's marginal value depends on what has already been picked (cf. Section 4), and they share one greedy template while differing in the objective that template grows against.

Clustering and coverage. The simplest objective spreads the selection across the data geometrically. MoDS, CaR, and DiverseEvol use k-center or clustering so the chosen samples cover the space and no region is over-represented Du et al. (2023); Ge et al. (2024); Wu et al. (2023), and DEITA adds a sample only when its representation brings genuinely new information through its Repr Filter Liu et al. (2024b).

Submodular and information-theoretic objectives. A more principled line writes the coverage goal as a monotone submodular function, whose diminishing-returns property guarantees that greedy selection is near-optimal. TAGCOS and DELIFT maximize a submodular coreset objective so the subset reproduces the behavior of the full data Zhang et al. (2024); Agarwal et al. (2025), QDIT maximizes a combined quality and facility-location score Bukharin et al. (2023), MIG maximizes information gain over a semantic label graph Chen et al. (2025), and ZIP greedily minimizes the subset's compression ratio as a model-free proxy for redundancy Yin et al. (2024).

Pros and Cons. This family optimizes coverage and curbs redundancy directly, which no single top-k cut can do, and the submodular formulation even comes with an approximation guarantee. The costs are that each greedy step must re-evaluate marginal gains over a large pool, which is expensive, and that the notion of coverage is only as trustworthy as the embedding or similarity space it is measured in.

6.5 Summary

The four families form a ladder. Moving from hard filtering to soft reweighting to optimization and finally to submodular subset selection, the operator grows steadily more complex, and in return it captures inter-sample interactions and approaches global optimality ever more faithfully (Table 5). The choice of rung correlates with

Table 5 Selection-policy families, ordered by increasing operator complexity and fidelity to inter-sample interactions.

Policy	Mechanism	Captures interactions?	Representative works
Hard filter (top-k/thresh.)	Sort by score, cut	No (per-sample)	IFD Li et al. (2024b), LESS Xia et al. (2024), MIWV Jiang et al. (2025), Selective DPO Gao et al. (2025), GRAPE Zhang et al. (2025a), AlpaGasus Chen et al. (2024)
Soft reweighting	Continuous per-sample weight/LR	Partially (via gradients)	ADAPT Zhao et al. (2026), Diversity-as-a-Reward Ling et al. (2025)
Optimization	Sequential or combinatorial optimization	Yes (within batch/round)	GREATS Wang et al. (2024), LEAD Lin et al. (2025), Instruction Mining Cao et al. (2023), CADS Wan et al. (2025)
Submodular subset	Greedy maximize coverage/info-gain	Yes (subset-relative)	MoDS Du et al. (2023), CaR Ge et al. (2024), DEITA Liu et al. (2024b), TAGCOS Zhang et al. (2024), DELIFT Agarwal et al. (2025), QDIT Bukharin et al. (2023), MIG Chen et al. (2025), ZIP Yin et al. (2024), DiverseEvol Wu et al. (2023)

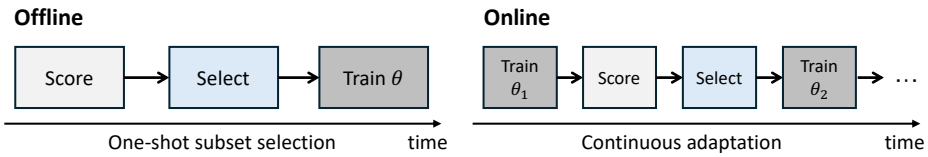

Figure 2 Offline vs. online data selection.

the rest of a method's design. Offline quality-only methods tend to use simple hard filtering, diversity-aware offline methods lean on coverage and submodular optimization, and online methods favor soft reweighting or greedy and bandit policies so they can adjust as training proceeds. The connecting principle is a simple one. The operator has to be sequential exactly when the reference is subset-relative (Section 4.3), because a per-sample score admits a one-shot top-k cut whereas a coverage objective forces greedy, one-sample-at-a-time growth.

7 Dimension 5: Selection Timing

The final dimension is timing, the question of *when* selection happens, and it is the watershed that runs through every method. It is also the primary split, offline versus online, reported in the rightmost column of Table 1. Timing settles two things that no other dimension can. It decides whether the selected subset is allowed to adapt to the model's continually evolving state, and whether selection can be computed once and reused. For this reason it is tightly coupled to the scoring-model choice of Section 5. Methods fall into the two regimes below, and a unifying compute-budget view, developed at the end of the section, cuts across both.

Two Selection-Timing Regimes

1. **Offline Selection** (Section 7.1): selection completes *once*, before training. Simple, reusable, and decoupled from the training pipeline, but fixed against a single model snapshot.
2. **Online Selection** (Section 7.2): selection is embedded *in* the training loop, including iterative self-evolution, and adapts to the model state at the cost of per-step scoring overhead.

7.1 Offline Selection

Offline methods complete all selection once, before training begins, and they make up the overwhelming majority of the field. It helps to notice that offline selection is not a single thing but covers two distinct arrangements that differ in how far the scoring model stands from the final one.

Two kinds of offline scoring. In the first, the scoring model is wholly separate from the target, as with the external-judge methods, the weak proxies, and human curation. No state of the target model is involved at all, so reuse is trivial but the proxy–target gap of Section 5 is at its widest. In the second, the score comes from a frozen, warmed-up snapshot θ_0 of the target model itself, as in offline self-scoring. This narrows the gap, since the scorer is at least the right model, but only at a single early state. Either way the appeal is the same combination of simplicity, a select-once and reuse-many workflow, and a clean decoupling of selection from training that eases deployment and reproducibility, which is why the influence and submodular coresets and the quality and diversity optimizers nearly all live here.

The cost of standing still. The shared limitation also follows from the decoupling. Training is non-stationary, and a sample's usefulness drifts as the model changes, a point made concrete by S2L's finding that loss *trajectories*, rather than single snapshots, carry the signal (Section 3.1). Because an offline subset is fixed against one scoring state, it is a static photograph of a moving target, so a sample that is informative early in training may add nothing later, and the selection has no way to react.

7.2 Online Selection and Iterative Self-Evolution

Online methods move selection inside the training loop, so the data used at each step is chosen in light of the model's current state θ_t. The family spans a range of coupling strengths. At the tight end, GREATS, LEAD, UDS, ADAPT, and Diversity-as-a-Reward select or reweight at essentially every step Wang et al. (2024); Lin et al. (2025); Zou et al. (2025); Zhao et al. (2026); Ling et al. (2025). At the loose end is *iterative self-evolution*, exemplified by DiverseEvol Wu et al. (2023), where the model is retrained in rounds and uses its own updated embeddings between rounds to grow the selected set, so that selection and training alternate rather than interleave at every step. The motivating claim, stated plainly by ADAPT and GREATS, is that selection synchronized with the training state generalizes better than any fixed subset can.

The engineering problem of in-loop scoring. The price of staying synchronized is that scoring now happens on the critical path of training, where a naive implementation would roughly double the cost by demanding an extra forward or backward pass per candidate. Almost every online method is therefore organized around avoiding that overhead. GREATS reuses quantities already computed in the backward pass through a ghost inner-product, UDS reuses logits it already has, LEAD maintains an exponential moving average of historical loss together with a bandit over clusters so that not every sample is scored afresh, and ADAPT folds its signal into the learning rate rather than into an explicit selection step. Seen this way, online selection is less a data-cleaning step than a co-design of the data and the optimizer, which places it close to curriculum and active learning. Its risks mirror its promise. Per-step greedy choices can be myopic, and a model that selects its own training data can reinforce its existing biases unless an explicit diversity or anchor term holds it in check.

7.3 A Unified Compute-Budget View

Compute-Constrained Data Selection Yin and Rush (2024) and CADS Wan et al. (2025) offer a view that cuts beneath the offline and online split. Their shared insight is that the compute spent on selection is not free and must be counted in the same budget as training, which reframes the whole problem. Compute-Constrained shows through a broad meta-analysis that, under a fixed total budget, the cheaper selection method is frequently the compute-optimal one, because a more accurate but expensive scorer can cost more than the training it was meant to improve. CADS makes the same commitment internally by writing the budget constraint into the inner level of its bilevel optimization. The offline and online regimes sit on opposite sides of this accounting. Offline scoring is paid once and then amortized over every run that reuses the subset, so its per-use cost falls the more it is reused, whereas online scoring is repaid in full on each run but buys an

Table 6 Offline vs. online selection along the axes that matter in practice.

Axis	Offline	Online (incl. self-evolution)
When	Once, before training	Every (few) step(s), in-loop
Adapts to model state	No (static snapshot)	Yes (uses θ_t)
Subset reuse	Yes (select once, reuse)	No (recomputed each run)
Selection overhead	Decoupled, paid once	Per-step, must avoid extra fwd/bwd
Generalization	Good, but state-mismatched	Better (training-synchronized)
Engineering focus	Pipeline simplicity, reproducibility	Cheap in-loop scoring (ghost dot-product, logits reuse)
Examples	AlpaGasus Chen et al. (2024), LESS Xia et al. (2024), DEITA Liu et al. (2024b), TAGCOS Zhang et al. (2024), QDIT Bukharin et al. (2023)	GREATS Wang et al. (2024), LEAD Lin et al. (2025), UDS Zou et al. (2025), ADAPT Zhao et al. (2026), Div.-as-Reward Ling et al. (2025)

adaptivity that no cached subset can. The real objective, then, is not to find the best subset in the abstract but to maximize the joint return of selection and training under a given total compute.

7.4 Trade-off Summary

The two regimes are mirror images of one another. Offline selection gives up contact with the evolving model state in exchange for simplicity, reusability, and zero online overhead, while online selection accepts per-step scoring overhead in exchange for adaptivity and better generalization (Table 6). Which trade is preferable depends on the compute budget, the stability of the target task, and how much the selected subset will be reused. Whichever route is taken, counting the selection overhead within the total budget should become standard practice when judging the quality of a method.

8 Open Challenges

Despite substantial progress, viewing the field through the five dimensions reveals several core challenges that remain unresolved.

8.1 The Cost–Fidelity Trade-off of Scoring

Scoring is rarely free, and its cost can offset the gains of selection. The most faithful metrics—gradient, influence, and Shapley signals—require per-sample gradients or repeated evaluation, while self-model online methods consume training compute at every step; cheap proxies (ICL, representation, weak-model loss) only approximate them (Section 3.7). This creates a two-fold difficulty: designing "cheap yet accurate" surrogates (ghost inner-products, low-rank approximations, logits reuse), and characterizing *when* a cheap proxy is good enough rather than discovering it empirically after the fact. The problem is sharpest at scale, where the per-step overhead of online selection may simply be unaffordable at billions or trillions of tokens—the key bottleneck to large-scale adoption.

8.2 Reliability and Bias of the Scoring Signal

Every scoring signal that depends on an external source inherits that source's limitations. Proxy-model methods (Dimension ③) face a *proxy–target gap*: data that is "good" on a small proxy is not necessarily beneficial to a large final model, and the gap widens with the scale difference. Anchor-comparison methods (Dimension ②) hinge on the quality of the external reference—a biased or incomplete anchor can cause targeted selection to *amplify* those biases, harming generalization and fairness. The external-LLM/reward-model family (Section 5.3) delivers strong, human-aligned signals but reflects *the judge's* notion of quality, inheriting its biases (verbosity, style, format) and raising API cost, reproducibility, and data-leakage concerns. Reducing

this dependence—reliable selection under weak or absent anchors, distilling judges into small open scorers, and calibrating signals against the target model—remains open.

8.3 Multi-Signal Fusion and the Limits of a Single Metric

A single metric (e.g., difficulty alone or influence alone) often misses something—scoring only for utility sacrifices diversity, while scoring only for diversity admits low-quality samples. The surveyed SFT methods make this concrete: almost all of them pair a quality score with an explicit diversity/coverage term (Section 3.6), and some fold both into a single objective. Yet how to weight quality against diversity, and how to fuse loss, gradient, distribution, and diversity signals in a principled rather than hand-tuned way, remains open—the balance is currently set by trial and error.

8.4 Evaluation and Reproducibility

Data selection methods lack a unified benchmark: different works use different candidate pools, target tasks, and compute budgets, making results hard to compare. Moreover, many methods' gains are sensitive to hyperparameters (threshold, k, subset size), challenging reproducibility. Establishing a unified evaluation protocol that accounts for selection compute is critically needed infrastructure.

8.5 The Lack of Theoretical Understanding

Most current methods are heuristic and lack rigorous theoretical characterization of *why* a given metric works. Influence functions and optimal transport provide partial theoretical lenses, but their applicability and tightness under the non-convex, large-scale training dynamics of LLMs remain insufficient. Connecting data selection to the theory of generalization and training dynamics is the long-term, fundamental challenge.

9 Conclusion

This survey proposed a unified five-dimensional taxonomy that decomposes the diverse landscape of LLM data selection into: ① the scoring metric, ② the comparison target, ③ the scoring model, ④ the selection policy, and ⑤ the selection timing. Using this framework, we studied representative works, characterizing each directly along the five dimensions and analyzing the design principles and trade-offs of each.

Looking ahead, we see the following directions as especially promising:

- *Selection for agentic data.* As LLMs are increasingly deployed as agents—reasoning and acting over tools and environments Yao et al. (2023); Schick et al. (2023)—the unit of selection shifts from static instruction–response pairs to entire interaction *trajectories* and tool-call traces. Such trajectories are typically *self-generated* and highly redundant: only a fraction are successful or informative, so filtering becomes essential rather than optional. Reasoning-bootstrapping methods already foreshadow this—STaR retains only the self-generated rationales that lead to correct answers and retrains on them Zelikman et al. (2022)—but they rely on a single correctness signal. Generalizing this into a principled selection step—scoring trajectories by task success, diversity of tool use, and informativeness for credit assignment, and pruning the large noisy rollout pools agents produce—is a largely open and promising direction, and a natural setting for the subset-relative, diversity-aware operators of Dimension ④.
- *Coupling with on-policy distillation.* On-policy distillation, in which the student learns from *its own* generations scored or relabeled by a teacher, has become a leading post-training recipe Agarwal et al. (2024); Gu et al. (2024), and at its core it is a selection problem: deciding which self-generated rollouts are worth distilling. Signals such as teacher–student disagreement, advantage estimates, and self-consistency Wang et al. (2023) are natural scores in this loop, tightly connecting the self-model regime (Dimension ③) and online timing (Dimension ⑤) with distillation, and giving data selection a central role in how models curate the data they themselves generate.
- *Unified benchmarks.* Progress is currently hard to measure because methods are evaluated on different candidate pools, target tasks, and budgets. The field needs a shared benchmark that fixes a common pool and downstream task suite, reports selection cost alongside final performance—so that "select once, train

once" baselines are comparable and the selection overhead is counted in the total budget Yin and Rush (2024); Wan et al. (2025)—and tests robustness to the hyperparameters (threshold, k, subset size) to which many methods are sensitive. Standardized leaderboards and released selected-subset artifacts would make results directly reproducible.

- *Deeper theory.* Most metrics remain heuristic, with no account of *why* they work. A predictive theory would characterize when a cheap proxy (e.g., representation similarity Ivison et al. (2025) or weak-model loss Li et al. (2024a); Yang et al. (2024)) provably tracks an expensive influence/Shapley signal Xia et al. (2024); He et al. (2024)—an empirical regularity already reported Yin and Rush (2024); Ivison et al. (2025)—and by how much it degrades otherwise; bound the proxy–target gap as a function of the scale difference; and connect a sample's selection score to its effect on generalization through the non-convex training dynamics of LLMs, moving beyond the partial lenses offered by influence functions Xia et al. (2024) and optimal transport Kang et al. (2024).

In sum, data selection is evolving from "empirical data cleaning" toward a principled training subsystem, and is increasingly entangled with how modern agents and post-training recipes generate their own data. We hope the five-dimensional framework proposed in this survey offers researchers a clear map for locating, comparing, and innovating within this design space.

References

Ishika Agarwal, Krishnateja Killamsetty, Lucian Popa, and Marina Danilevsky. DELIFT: Data efficient language model instruction fine-tuning. In *Proceedings of ICLR*, 2025.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *Proceedings of ICLR*, 2024.

Alexander Bukharin, Shiyang Li, Zhengyang Wang, Jingfeng Yang, Bing Yin, Xian Li, Chao Zhang, Tuo Zhao, and Haoming Jiang. Data diversity matters for robust instruction tuning, 2023.

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. Instruction mining: Instruction data selection for tuning large language models, 2023.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. AlpaGasus: Training a better alpaca with fewer data. In *Proceedings of ICLR*, 2024.

Yicheng Chen, Yining Li, Kai Hu, Zerun Ma, Haochen Ye, and Kai Chen. MIG: Automatic data selection for instruction tuning by maximizing information gain in semantic space. In *Findings of ACL*, 2025.

Xun Deng, Han Zhong, Rui Ai, Fuli Feng, Zheng Wang, and Xiangnan He. Less is more: Improving LLM alignment via preference data selection, 2025.

Qianlong Du, Chengqing Zong, and Jiajun Zhang. MoDS: Model-oriented data selection for instruction tuning, 2023.

Chengqian Gao, Haonan Li, Liu Liu, Zeke Xie, Peilin Zhao, and Zhiqiang Xu. Principled data selection for alignment: The hidden risks of difficult examples, 2025.

Yuan Ge, Yilun Liu, Chi Hu, Weibin Meng, Shimin Tao, Xiaofeng Zhao, Hongxia Ma, Li Zhang, Boxing Chen, Hao Yang, Bei Li, Tong Xiao, and Jingbo Zhu. Clustering and ranking: Diversity-preserved instruction selection through expert-aligned quality estimation. In *Proceedings of EMNLP*, 2024.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *Proceedings of ICLR*, 2024.

Yexiao He, Ziyao Wang, Zheyu Shen, Guoheng Sun, Yucong Dai, Yongkai Wu, Hongyi Wang, and Ang Li. SHED: Shapley-based automated dataset refinement for instruction fine-tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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

Tingyu Jiang, Shen Li, Yiyao Song, Lan Zhang, Hualei Zhu, Yuan Zhao, Xiaohang Xu, Kenjiro Taura, and Hao Henry Wang. Importance-aware data selection for efficient LLM instruction tuning, 2025.

Feiyang Kang, Hoang Anh Just, Yifan Sun, Himanshu Jahagirdar, Yuanzhi Zhang, Rongxing Du, Anit Kumar Sahu, and Ruoxi Jia. Get more for less: Principled data selection for warming up fine-tuning in LLMs, 2024.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In *Proceedings of ACL*, 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 NAACL*, 2024b.

Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Ling-Hao Chen, Junhao Liu, Tongliang Liu, Fei Huang, and Yongbin Li. One-shot learning as instruction data prospector for large language models. In *Proceedings of ACL*, 2024c.

Zhuang Li, Yuncheng Hua, Thuy-Trang Vu, Haolan Zhan, Lizhen Qu, and Gholamreza Haffari. SCAR: Data selection via style-consistency-aware response ranking for efficient instruction-tuning of large language models. In *Proceedings of ACL*, 2025.

Xiaotian Lin, Yanlin Qi, Yizhang Zhu, Themis Palpanas, Chengliang Chai, Nan Tang, and Yuyu Luo. LEAD: Iterative data selection for efficient LLM instruction tuning. In *Proceedings of the VLDB Endowment*, 2025.

Zhenqing Ling, Daoyuan Chen, Liuyi Yao, Qianli Shen, Yaliang Li, and Ying Shen. Diversity as a reward: Fine-tuning LLMs on a mixture of domain-undetermined data, 2025.

Liangxin Liu, Xuebo Liu, Derek F. Wong, Dongfang Li, Ziyi Wang, Baotian Hu, and Min Zhang. SelectIT: Selective instruction tuning for large language models via uncertainty-aware self-reflection, 2024a.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *Proceedings of ICLR*, 2024b.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. #InsTag: Instruction tagging for analyzing supervised fine-tuning of large language models. In *Proceedings of ICLR*, 2024.

Xingyuan Pan, Luyang Huang, Liyan Kang, Zhicheng Liu, Yu Lu, and Shanbo Cheng. G-DIG: Towards gradient-based diverse and high-quality instruction data selection for machine translation. In *Proceedings of ACL*, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Weilin Wan, Weizhong Zhang, and Cheng Jin. Computational budget should be considered in data selection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Jiachen T. Wang, Tong Wu, Dawn Song, Prateek Mittal, and Ruoxi Jia. GREATS: Online selection of high-quality data for LLM training in every iteration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. https://openreview.net/forum?id=232VcN8tSx.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of ICLR*, 2023.

Shengguang Wu, Keming Lu, Benfeng Xu, Junyang Lin, Qi Su, and Chang Zhou. Self-evolved diverse data sampling for efficient instruction tuning, 2023.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of ICML*, 2024.

Yu Yang, Siddhartha Mishra, Jeffrey N. Chiang, and Baharan Mirzasoleiman. SmallToLarge (S2L): Scalable data selection for fine-tuning large language models by summarizing training trajectories of small models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *Proceedings of ICLR*, 2023.

Junjie Oscar Yin and Alexander M. Rush. Compute-constrained data selection, 2024.

Mingjia Yin, Chuhan Wu, Yufei Wang, Hao Wang, Wei Guo, Yasheng Wang, Yong Liu, Ruiming Tang, Defu Lian, and Enhong Chen. Entropy law: The story behind data compression and LLM performance. In *Findings of EMNLP*, 2024.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Dylan Zhang, Qirun Dai, and Hao Peng. The best instruction-tuning data are those that fit, 2025a.

Hong Zhang, Feng Zhao, Ruilin Zhao, Cheng Yan, and Kangzheng Liu. Priority on high-quality: Selecting instruction data via consistency verification of noise injection. In *Proceedings of EMNLP*, 2025b. https://aclanthology.org/2025.emnlp-main.1048/.

Jipeng Zhang, Yaxuan Qin, Renjie Pi, Weizhong Zhang, Rui Pan, and Tong Zhang. TAGCOS: Task-agnostic gradient clustered coreset selection for instruction tuning data, 2024.

Wanru Zhao, Hongxiang Fan, Shell Xu Hu, Wangchunshu Zhou, Bofan Chen, and Nicholas D. Lane. CLUES: Collaborative private-domain high-quality data selection for LLMs via training dynamics. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Wanru Zhao, Yihong Chen, Yuzhi Tang, Wentao Ma, Shengchao Hu, Shell Xu Hu, Alex Iacob, Abhinav Mehrotra, and Nicholas D. Lane. Rethinking data curation in LLM training: Online reweighting offers better generalization than offline methods, 2026.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Heming Zou, Yixiu Mao, Yun Qu, Qi Wang, and Xiangyang Ji. Utility-diversity aware online batch selection for LLM supervised fine-tuning, 2025.