PRINCIPAL-COMPONENT CURRICULUM LEARNING FOR MULTI-SOURCE KNOWLEDGE DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Curriculum learning, which organizes training data from easy to hard, is increasingly used in knowledge distillation for large language models. Existing distillation curricula rank examples by estimated difficulty and thus decide when each example enters training. In practice, however, distillation data are mixtures drawn from multiple sources, and each update combines learning signals that differ in task, domain, and format. A curriculum for such data must therefore answer two questions: which examples the student learns first, and which it learns together. We propose principal-component curriculum learning (PCCL) to address both. PCCL orders batches by a weighted principal-loading score derived from per-example teacher gradients, so that the student first encounters the pool's dominant learning signal before more specialized ones. For multi-source pools, PCCL clusters examples with character n-gram TF–IDF features and KMeans and builds batches that mix examples across domains, so that each update draws on multiple sources. Experiments on single-source and multi-source distillation with Qwen and Llama students show that PCCL consistently improves ROUGE-L over random batching and reaches strong performance earlier in training.

1 INTRODUCTION

Knowledge distillation transfers the capabilities of large language models to smaller students (Hinton et al., 2015; Gu et al., 2024). In practice, distillation data are rarely homogeneous: large-scale distillation and post-training draw on broad mixtures spanning different tasks, domains, and response formats. Gemma 2, for instance, trains its smaller models with knowledge distillation and combines synthetic and human-written data from internal and public sources during post-training (Team et al., 2024). Curriculum learning offers a way to organize how the student encounters such supervision, conventionally by ordering examples from easy to hard according to estimated difficulty, although difficulty ordering alone often yields limited gains (Bengio et al., 2009; Platanios et al., 2019; Wu et al., 2020). Recent distillation methods apply this principle to multi-source data: TAPIR combines curriculum planning with task rebalancing and response refinement (Yue et al., 2024), and CLPD pairs an easy-to-hard data curriculum with teachers of increasing capacity (Cao et al., 2026). These methods decide *when* each example enters training.

Mixed data, however, raise a second question that difficulty ordering leaves open: which examples the student learns *together*. Examples in the same batch jointly determine each parameter update, and in a multi-source pool they carry signals that differ in task, format, and learning objective, which may reinforce or interfere with one another. Difficulty ordering shapes batch composition only implicitly, grouping examples by difficulty rather than by their relationships. Figure 1 illustrates this limitation: a heterogeneous pool may contain distinct groups of related examples, but a scalar difficulty score collapses them into one ordering and says nothing about which examples should be learned together. This motivates curriculum design around two questions: which examples should the student learn first, and which should it learn together?

To address these two questions, we introduce principal-component curriculum learning (PCCL), a framework that derives curriculum decisions from relationships among per-example teacher gradients. PCCL represents each example by its position in a compact relational space derived from the frozen teacher's per-example gradients, rather than by difficulty alone, and scores it by a weighted combination of its loadings on selected principal directions. Examples with high scores align with

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

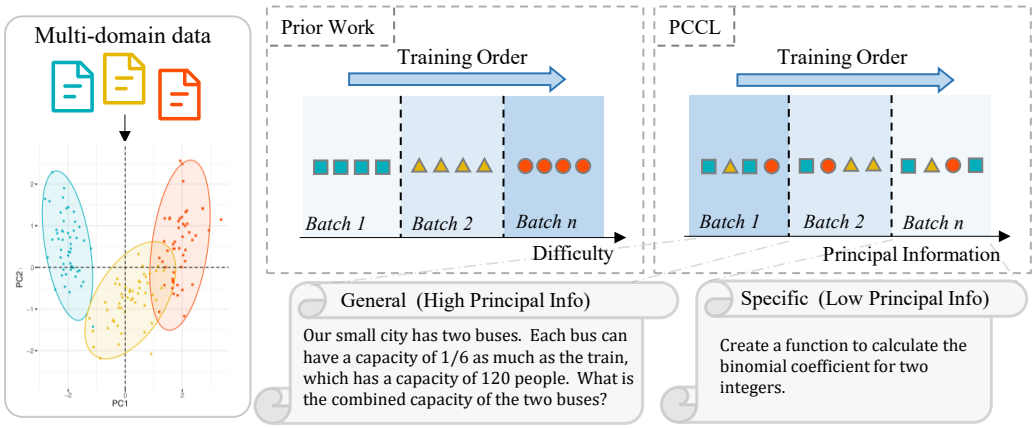


Figure 1: Difficulty-based curricula collapse heterogeneous examples into a scalar ordering, potentially segregating different sources across batches and overlooking which examples should be learned together. PCCL prioritizes examples with higher principal information while preserving cross-source diversity within each batch. Examples with higher principal information tend to capture more general patterns, whereas lower-scoring examples are more specialized.

the dominant directions of the teacher's relational structure, so presenting high-scoring batches first exposes the student to the pool's dominant learning signal before more specialized ones. This score determines which examples the student learns first. To determine which examples it learns together, PCCL adapts batch construction to the heterogeneity of the training pool. In single-source pools, batches are formed at random and then ordered by the score, isolating the effect of the teacher-derived order. In multi-source pools, PCCL first clusters examples using character n-gram TF–IDF features and KMeans, and builds batches that draw from different clusters so that examples from different domains contribute to the same update; the same score then orders these batches. PCCL thus uses a single teacher-derived signal to set the order, while adapting batch membership to the heterogeneity of the pool.

- We formulate curriculum design for distillation around two questions: which examples the student learns first and which it learns together. To address them, we introduce principal-component curriculum learning (PCCL), which orders training by relationships among per-example teacher gradients and adapts batch composition to the heterogeneity of the training pool.
- We propose a weighted principal-loading score that ranks batches by their alignment with the dominant principal directions of the teacher's gradient structure. For multi-source pools, we combine this score with character n-gram TF–IDF features and KMeans clustering to build batches that mix examples across domains before ordering them.
- We evaluate PCCL on single-source and multi-source distillation with Qwen and Llama students. Compared with random batching, PCCL improves ROUGE-L by 2.16 points on the single-source pool and by 0.78 and 0.85 points on the multi-source pool for Qwen and Llama, respectively. PCCL reaches approximately 43.5 ROUGE-L at 300 optimizer steps, whereas random batching reaches a comparable score at 350 steps, corresponding to 14.3% fewer steps and faster convergence in addition to higher final scores.

2 Related Work

Curricula and distillation. Curriculum learning organizes training exposure through sample difficulty and learning progress (Bengio et al., 2009; Platanios et al., 2019). Self-paced learning jointly updates sample selection and model parameters (Kumar et al., 2010), while transfer-based curricula estimate difficulty using classifiers built on pretrained representations (Weinshall et al., 2018). Its benefits vary across training regimes (Wu et al., 2020); for LLM mathematical reasoning, the

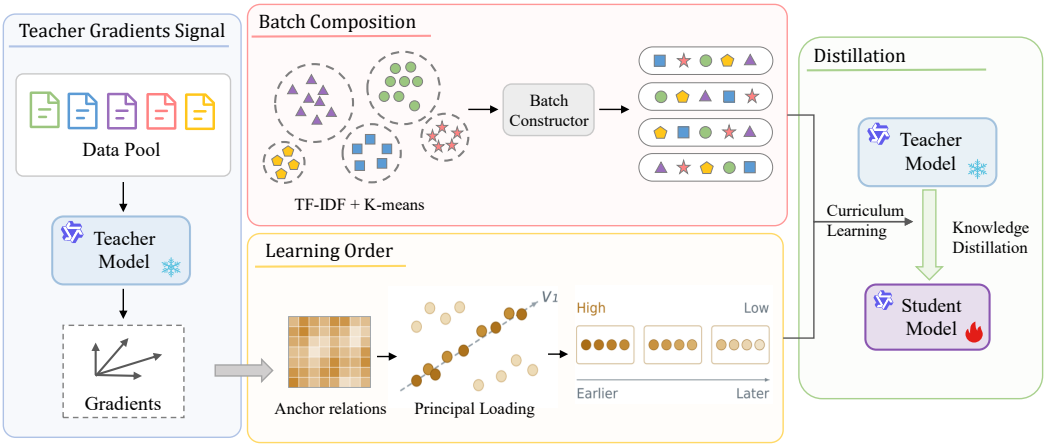


Figure 2: Overview of PCCL. Teacher-gradient relations define a principal-loading curriculum that determines the learning order, while text-derived groups guide cross-source batch construction in heterogeneous settings. PCCL therefore coordinates when training signals are introduced with which signals are learned together within each update.

preferred ordering also depends on the model, task, and difficulty metric (Jia et al., 2026). In instruction distillation, TAPIR combines curriculum planning with task rebalancing and response refinement (Yue et al., 2024), while CLPD couples easy-to-hard ordering with teachers of increasing capability (Cao et al., 2026). Sequence-level distillation trains students on beam-decoded teacher sequences (Kim & Rush, 2016). Other distillation methods change the supervision itself: MiniLLM uses reverse-KL optimization (Gu et al., 2024), generalized knowledge distillation learns from teacher feedback on student-generated sequences (Agarwal et al., 2024), and relational knowledge distillation matches distances and angles among teacher and student representations (Park et al., 2019). Data-mixture methods provide another way to control exposure: DoReMi learns domain weights using a proxy model, and RegMix predicts effective mixtures from small-model training runs (Xie et al., 2023; Liu et al., 2025).

Gradient-based data organization. Gradient-based methods use information about how examples affect model updates to estimate their influence, select training subsets, and organize data order and batch composition. TracIn estimates training-example influence through gradient inner products across model checkpoints (Pruthi et al., 2020). Gradient representations underpin scalable attribution in TRAK (Park et al., 2023), target-task instruction selection in LESS (Xia et al., 2024), task-agnostic coreset selection in TAGCOS (Zhang et al., 2025), and graph-based instruction selection in G2IS (Zhao et al., 2025). Determinantal point processes over normalized gradient representations also support diverse instruction subset selection (Wang et al., 2024b). CRAIG constructs weighted coresets through submodular optimization to approximate the full gradient (Mirzasoleiman et al., 2020), while GRAD-MATCH selects weighted subsets to approximate training- or validation-set gradients (Killamsetty et al., 2021). Beyond subset selection, GraB constructs full-data permutations through gradient balancing (Lu et al., 2022), while stratified SGD and Gradient Clustering use cluster structure to reduce gradient estimation variance (Zhao & Zhang, 2014; Faghri et al., 2020). Batch membership is also an explicit design choice in contrastive learning: spectral clustering constructs high-loss batches (Cho et al., 2023), and JEST jointly selects examples from larger candidate batches (Evans et al., 2024). For language models, GREATS selects batches online using an approximation of validation-loss improvement (Wang et al., 2024a), and PartitionSel couples validation-guided gradient matching with per-domain selection budgets (Agrawal et al., 2026). These methods establish precedents for gradient relations, ordering, and batch construction.

3 METHOD

PCCL structures curriculum learning along two complementary axes: when training signals are introduced and which signals are combined within each update. To determine the learning order,

Algorithm 1 Principal-component curriculum learning (PCCL)

Require: pool $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$; teacher T; blocks L; sketch width d; anchors m; directions q, weights w; batch size b; groups r; seed

Stage 1 – relational representation and principal loading
$\Theta \leftarrow$ q/k/v/o weights of the last L blocks of T; $S \leftarrow$ CountSketch $\mathbb{R}^{|\Theta|} \to \mathbb{R}^d$
$g_i \leftarrow S\,\nabla_\Theta\big[-\log p_T(y_i \mid x_i)\big]$ for $i = 1, \dots, n$ {per example; unnormalised}
$A \leftarrow m$ anchors drawn without replacement from $[n]$; $R_{ij} \leftarrow \langle g_i, g_{a_j} \rangle$
$\Phi \leftarrow H_n R H_m$, $H_p = I_p - \frac{1}{p}\mathbf{1}\mathbf{1}^\top$; $\Phi = U\Sigma V^\top$, $V_q \leftarrow [v_1, \dots, v_q]$
$\lambda(i) \leftarrow \big(\sum_{j \le q} w_j |\phi_i v_j|^2 / \|\phi_i\|^2\big)^{1/2}$ with $\phi_i = \Phi_{i,:}$
Stage 2 – batch construction and ordering
if single-source **then**
  $\mathcal{B} \leftarrow$ one shuffle of $\mathcal{D}$, cut into batches of size b
else
  $\{C_c\}_{c=1}^{r} \leftarrow k$-means on character n-gram TF–IDF of instruction $\|$ response; shuffle each C_c
  while $\bigcup_c C_c \neq \emptyset$ **do**
    $M \leftarrow \emptyset$; **repeat** pop one example from each of the $\le b - |M|$ largest nonempty groups into M **until** $|M| = b$ or all groups are empty
    append M to $\mathcal{B}$
  end while
end if
$s(M) \leftarrow |M|^{-1} \sum_{i \in M} \lambda(i)$ for each $M \in \mathcal{B}$; sort $\mathcal{B}$ by decreasing $s(M)$
return $\mathcal{B}$, replayed in this order every epoch

we use a frozen teacher to build a relational representation from per-example gradients and extract its dominant principal directions. Examples with larger principal loadings are prioritized, so the student first learns from signals that better reflect the shared structure of the training pool before moving toward more specialized patterns. In multi-source settings, however, priority alone does not control what is learned together. PCCL therefore complements the ordering signal with text-based grouping and constructs batches across different task and format regions, encouraging each update to integrate supervision from multiple parts of the pool. The two components play distinct but coordinated roles: principal loading determines when examples are emphasized, while batch construction determines which examples interact within an update. Figure 2 illustrates the overall framework, and Algorithm 1 summarizes the full procedure.

3.1 Preliminaries

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ be the distillation pool, T a teacher with parameters θ_T, and f_ϕ the student. For an example (x_i, y_i), the student is trained with the response-masked forward-KL objective at temperature $T_{\mathrm{KD}} = 2$:

$$\ell_{\mathrm{KD},i}(\phi; \theta_T) = \mathrm{KL}\Big(p_T^{(2)}(\cdot \mid x_i, y_i) \,\Big\|\, p_\phi^{(2)}(\cdot \mid x_i)\Big), \tag{1}$$

where $p_T^{(2)}$ and $p_\phi^{(2)}$ are the teacher and student token distributions after dividing logits by T_{KD}. For an optimizer batch B_t, the student loss and update direction are

$$\mathcal{L}_{B_t}(\phi) = \frac{1}{|B_t|} \sum_{i \in B_t} \ell_{\mathrm{KD},i}(\phi; \theta_T), \qquad u_t = \nabla_\phi \mathcal{L}_{B_t}(\phi_t). \tag{2}$$

Thus, batch membership determines which learning signals are combined, whereas the order of the batches determines when those signals affect the student. PCCL keeps these roles explicit: teacher relations provide the ordering signal, and multi-source text groups provide the additional membership signal.

3.2 Teacher Gradients as a Curriculum Signal

Difficulty is an example-level attribute; it does not reveal whether two examples induce compatible or conflicting local updates. For the teacher loss ℓ_T, a positive gradient inner product means that

a small teacher update that reduces one example's loss also tends to reduce the other's loss to first order, while a negative inner product indicates local conflict. We use these relations as a teacher-side prior for organizing the student's supervision. The prior is useful when the teacher and student are trained toward the same targets, but it is not assumed to be an exact representation of the student's changing gradients; the student-side effect is evaluated empirically.

We compute the signal once from the frozen teacher. To obtain a compact, reproducible representation, θ_T^{sel} contains the query, key, value, and output projections in the last four attention blocks. These projections capture attention-mediated token interactions while keeping extraction and storage manageable; the MLP projections are excluded to keep the selected parameter family fixed across models. For each example, we sketch the selected teacher gradient with a shared CountSketch map:

$$h_i = \nabla_{\theta_T^{\mathrm{sel}}} \ell_T(x_i, y_i; \theta_T), \qquad g_i = S h_i \in \mathbb{R}^d, \quad d = 2048. \tag{3}$$

Signed hashing gives an unbiased estimator of gradient inner products over the sketch randomness, with error controlled by the sketch dimension and gradient norms (Weinberger et al., 2009; Larsen et al., 2021). The sketch is therefore an efficient approximation to the selected teacher geometry, rather than a claim that the full gradient is preserved exactly.

To express each example relative to a shared reference set, we select $m = 256$ anchors uniformly without replacement and form

$$R_{ij} = \langle g_i, g_{a_j} \rangle, \qquad K = H_n R H_m, \qquad H_n = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top, \quad H_m = I_m - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^\top. \tag{4}$$

The row $k_i = K_{i,:}$ is the relational signature of example i. Double centering removes global row and anchor offsets, so subsequent spectral analysis captures relative structure rather than raw gradient magnitude. The signatures and all derived scores are computed once and reused across student runs.

3.3 Choosing What the Student Learns First

Let $K = U\Sigma V^\top$ be its singular value decomposition, and let v_1 be the leading right singular direction. We define the principal-loading score

$$\lambda(i) = \frac{|k_i v_1|}{\|k_i\|_2}, \tag{5}$$

with $\lambda(i) = 0$ for a zero signature. This is the normalized amount of an example's relational energy aligned with the pool's dominant axis. The absolute value makes the score invariant to the arbitrary sign of v_1. The formulation extends directly to a weighted sum of several singular directions; our experiments use the first direction with unit weight, so the reported score has no additional weighting hyperparameter.

The leading direction captures the strongest shared variation in the teacher's relational signatures. Presenting high-loading batches first therefore exposes the student to the pool's most common relational pattern before more specialized deviations. This is a curriculum prior rather than an assertion that high-loading examples are easier. After batch membership is constructed, we score a batch by

$$s(B) = \frac{1}{|B|}\sum_{i \in B} \lambda(i) \tag{6}$$

and sort batches in decreasing $s(B)$, with seeded tie-breaking. In the single-source setting, the batches are formed by a random partition of the pool; the loading score then supplies the only curriculum intervention.

3.4 Choosing What the Student Learns Together

In a multi-source pool, two examples can have similar loading while belonging to different task or response-format regions. The membership rule should therefore expose these regions within an update rather than rely on a scalar priority to discover them. We concatenate each instruction and response, represent the text with character TF–IDF features using 3–5-grams, at most 30,000 features, and minimum document frequency 3, and apply KMeans (Lloyd, 1982) with $r = 4$ groups and 10 initializations. This label-free representation is chosen because it makes task and format

Table 1: Multi-source distillation with Qwen3-8B → Qwen3-0.6B. Scores are ROUGE-L (×100, mean ± SD over five seeds); All is the pooled mean over evaluation examples.

Method	SQuAD	SamSum	GSM8K	SelfInst	Vicuna	S-NI	UnNI	All
Composition baselines								
DPP	69.8 ±1.2	38.9 ±0.9	**44.0 ±0.7**	20.0 ±1.1	18.3 ±1.2	38.2 ±0.7	39.7 ±1.4	43.08 ±0.52
Ordering baselines								
FCL	68.2 ±1.1	36.2 ±0.7	41.6 ±0.6	18.8 ±0.7	19.8 ±0.6	36.3 ±0.4	39.9 ±0.6	41.55 ±0.49
RCL	67.2 ±1.8	38.7 ±0.3	40.2 ±0.9	19.6 ±0.7	14.4 ±0.8	33.6 ±1.3	34.9 ±2.0	40.11 ±0.91
GFC	68.8 ±0.4	38.6 ±0.2	42.8 ±0.6	18.5 ±0.7	17.6 ±0.8	37.1 ±0.8	39.8 ±1.5	42.32 ±0.63
GRC	66.4 ±2.4	38.7 ±0.3	42.6 ±1.0	19.7 ±1.2	16.3 ±0.8	37.8 ±0.6	38.0 ±1.9	41.75 ±0.62
CLPD	70.5 ±0.5	37.3 ±0.5	40.1 ±0.9	20.0 ±0.6	15.5 ±1.0	35.2 ±0.9	37.3 ±0.4	41.22 ±0.24
IFD-order	65.7 ±1.7	37.2 ±1.2	39.3 ±0.4	17.5 ±0.5	10.8 ±0.6	30.6 ±0.9	35.4 ±2.6	38.69 ±0.61
GraB	69.5 ±1.2	39.0 ±0.8	42.8 ±0.8	20.0 ±1.5	18.9 ±1.0	38.1 ±0.7	39.5 ±0.9	42.81 ±0.48
CORGI	70.5 ±0.9	**39.5 ±0.5**	42.4 ±0.4	19.5 ±0.8	18.7 ±1.0	37.2 ±0.9	38.7 ±1.3	42.65 ±0.43
DUCL	70.1 ±1.0	38.2 ±0.3	43.0 ±0.6	19.4 ±0.4	**20.1 ±0.5**	37.3 ±0.4	39.4 ±0.9	42.64 ±0.39
Loss schedule								
CTKD	68.3 ±1.8	38.3 ±0.2	42.4 ±0.5	19.3 ±0.9	16.6 ±0.8	36.8 ±1.2	39.5 ±2.5	42.05 ±0.49
random	70.0 ±1.0	38.9 ±0.4	43.9 ±0.6	19.9 ±0.8	17.7 ±1.0	37.8 ±1.1	39.7 ±0.6	43.05 ±0.35
PCCL	**71.8 ±1.2**	39.0 ±0.4	43.7 ±0.4	**20.4 ±0.4**	18.7 ±0.3	**38.6 ±0.8**	**41.3 ±0.8**	**43.83 ±0.27**

cues explicit; the teacher-relational signature is used for ordering, not treated as a human-readable domain label.

For each group, we shuffle its members once with a fixed seed. We then fill each batch by taking one example from as many nonempty groups as the available slots permit, prioritizing groups with the largest remaining counts and repeating rounds when slots remain. This produces a partition of the pool with cross-group membership. We score the resulting batches with Eq. 6 and present them in that order. The same construction is reused in each epoch, and the complete procedure is described in Algorithm 1.

4 Experiments

We evaluate PCCL in both single-source and multi-source distillation settings to study how curriculum design behaves under different levels of data heterogeneity. The experiments focus on the two complementary decisions at the core of PCCL: how training signals are ordered over the course of learning and how they are combined within each update. We compare against representative ordering and batch-composition baselines and further disentangle the contribution of each component through ablation studies.

4.1 Experimental Settings

4.1.1 Datasets

We construct two training pools of $4{,}000$ examples. The single-source setting uses examples from Dolly-15k (Conover et al., 2023), a human-written instruction dataset covering tasks such as question answering, information extraction, and summarization. The multi-source setting combines $1{,}000$ examples from each of SQuAD (Rajpurkar et al., 2016), SAMSum (Gliwa et al., 2019), CodeAlpaca (Chaudhary, 2023), and GSM8K (Cobbe et al., 2021), covering reading comprehension, dialogue summarization, code generation, and mathematical reasoning, respectively. The two settings let us study organization within one instruction collection and across collections with different task formats. Appendix A.1 provides a dataset-by-dataset description of the training and evaluation corpora.

We evaluate both in-distribution (ID) performance and transfer to out-of-distribution (OOD) benchmarks. For the single-source setting, DollyEval provides 500 ID examples. For the multi-source setting, the ID evaluation includes SQuAD, SAMSum, and GSM8K. For OOD evaluation, we use

Table 2: Single-source distillation with Qwen3-8B → Qwen3-0.6B. Scores are ROUGE-L (×100, mean ± SD over five seeds); All is the pooled mean over evaluation examples.

Method	DollyEval	SelfInst	Vicuna	S-NI	UnNI	All
Composition baselines						
DPP	23.7 ±0.3	18.9 ±0.8	18.7 ±0.6	31.2 ±1.5	30.2 ±1.0	26.69 ±0.64
Ordering baselines						
FCL	22.1 ±0.4	17.2 ±0.5	18.3 ±0.3	28.6 ±0.5	26.0 ±0.6	24.12 ±0.27
RCL	21.9 ±0.4	**19.7 ±0.7**	15.5 ±0.4	33.6 ±0.7	**32.8 ±0.9**	27.50 ±0.52
GFC	21.8 ±0.2	17.0 ±0.6	19.4 ±0.3	27.8 ±0.7	26.2 ±0.5	23.92 ±0.35
GRC	23.0 ±0.8	18.1 ±1.2	18.7 ±0.8	31.0 ±1.3	29.5 ±1.1	26.12 ±0.73
CLPD	22.0 ±0.3	18.8 ±0.7	17.8 ±0.3	31.1 ±0.7	29.8 ±0.6	26.03 ±0.44
IFD-order	**23.9 ±0.2**	17.9 ±0.6	16.9 ±1.3	32.0 ±0.9	32.4 ±1.4	27.37 ±0.60
GraB	23.4 ±0.2	18.7 ±0.6	18.9 ±0.5	31.2 ±0.4	30.0 ±0.8	26.52 ±0.20
CORGI	22.2 ±0.2	18.7 ±0.8	**19.5 ±0.2**	31.7 ±0.4	29.5 ±0.3	26.25 ±0.26
DUCL	22.9 ±0.4	18.9 ±0.4	19.4 ±0.2	30.6 ±0.6	28.3 ±0.5	25.80 ±0.28
Loss schedule						
CTKD	22.5 ±1.1	17.7 ±0.8	18.3 ±0.5	29.4 ±0.7	28.3 ±0.4	25.16 ±0.26
random	22.9 ±0.7	18.6 ±0.2	18.9 ±0.9	29.9 ±1.5	29.6 ±1.6	25.89 ±0.94
PCCL	23.6 ±0.4	18.3 ±0.5	19.3 ±0.3	**34.1 ±0.9**	32.6 ±0.5	**28.05 ±0.40**

SelfInst (Wang et al., 2023), Vicuna (Chiang et al., 2023), Super-NaturalInstructions (S-NI) (Wang et al., 2022), and Unnatural Instructions (UnNI) (Honovich et al., 2023).

4.1.2 Baselines

We compare composition methods based on diversity and ordering methods based on difficulty, interleaving, or gradient balancing. Representative comparisons include DPP diversification (Zhang et al., 2017), CORGI (Lee et al., 2024), DUCL (Jiang et al., 2026), and GraB (Lu et al., 2022); CTKD (Li et al., 2023) is included as a separate temperature-schedule reference. Appendix A.2 provides a detailed description of these baseline methods.

4.1.3 Evaluation Metrics

Following the evaluation convention used for large-scale instruction following (Gu et al., 2024), we use ROUGE-L scores on the validation set to select hyper-parameters. ROUGE-L (Lin, 2004) is based on the longest common subsequence and has been shown to align better with human preference than validation loss for this setting (Wang et al., 2022). We report ROUGE-L separately for each downstream benchmark and use the pooled example-level mean for the aggregate column. We also report task-specific metrics for the checkable benchmarks: token-level F1 on SQuAD and answer accuracy on GSM8K and StrategyQA. These scores are interpreted within their respective task columns and are not compared in absolute value across different metrics.

4.1.4 Implementation Details

The main-text experiments use a Qwen3-8B teacher and a Qwen3-0.6B student (Yang et al., 2025); the Llama replication is reported in Appendix B. For each setting, we first supervised-fine-tune the teacher on the corresponding $4{,}000$-example pool and then freeze it during student training. The student is optimized with token-level forward-KL distillation at temperature $T = 2$. Each run lasts three epochs with a context length of 512 tokens and a batch size of 64. The student optimizer is AdamW with weight decay 0.01, a learning rate of 2×10^{-5}, and global gradient clipping at 1.0.

4.2 Main Results

Tables 1 and 2 present the Qwen results for the single-source and multi-source settings. PCCL orders randomly constructed batches in the single-source setting; in the multi-source setting, it first constructs batches across character n-gram TF–IDF KMeans groups and then orders them by mean principal loading. We compare these setting-specific policies with composition, difficulty-based, and

Table 3: Downstream results across three different tasks. SQuAD is scored with token F1 and GSM8K and StrategyQA with accuracy.

Method	Qwen			Llama	
	SQuAD	GSM8K	StrategyQA	SQuAD	GSM8K
DPP	70.76	46.48	49.05	62.19	**45.08**
FCL	69.67	45.84	50.12	63.66	43.04
RCL	69.45	45.36	49.49	64.01	43.08
GFC	70.58	45.56	49.49	65.73	43.60
GRC	69.67	**47.04**	48.47	65.00	43.84
CLPD	71.41	41.80	51.14	62.27	42.52
IFD-order	66.00	38.04	48.96	55.40	37.08
GraB	72.57	45.76	49.30	65.87	43.16
CORGI	71.71	45.16	49.15	63.92	44.12
DUCL	71.40	44.80	50.27	64.14	42.88
random	70.83	45.52	50.75	65.48	43.36
PCCL	**72.59**	45.48	**51.72**	**67.17**	44.04

Table 4: Ablation of batch composition and batch ordering for multi-source distillation with Qwen3-8B $\rightarrow$ Qwen3-0.6B. Values are ROUGE-L scores on the seven evaluation benchmarks.

Method	SQuAD	SamSum	GSM8K	SelfInst	Vicuna	S-NI	UnNI
random	70.0	38.9	43.9	19.9	17.7	37.8	39.7
w/o ordering	71.5	39.1	43.9	19.9	18.2	38.2	40.9
w/o batch composition	70.5	38.9	**44.0**	20.2	18.3	38.0	40.6
Full PCCL	**71.8**	39.0	43.7	**20.4**	**18.7**	**38.6**	**41.3**

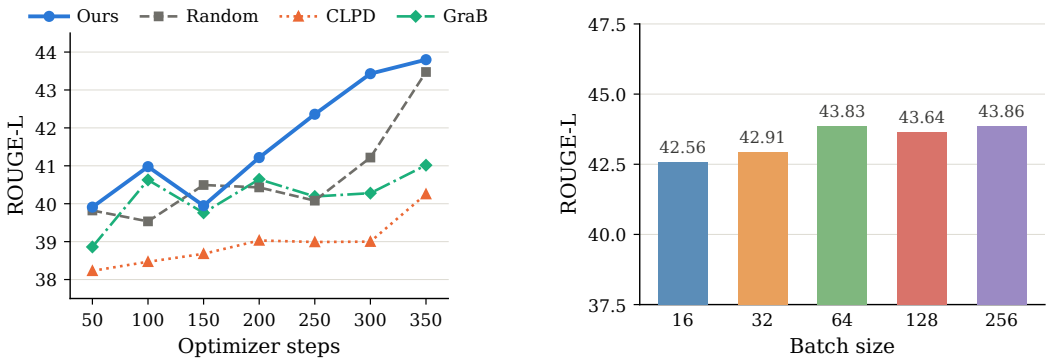
gradient-based curricula. The Llama replication is presented in Appendix B. In table 3, we report the corresponding task-level results using F1 for SQuAD and accuracy for GSM8K and StrategyQA, covering both Qwen and Llama students.

From the experiment result, we can see that PCCL improves aggregate performance in both settings. On the single-source Dolly pool, PCCL obtains the highest pooled ROUGE-L score, 28.05, compared with 25.89 for random and 27.50 for the strongest retained ordering baseline, RCL. The 2.16-point gain shows that a teacher-derived order can remain useful even when batch membership is otherwise random, where there are no source labels to exploit. On the multi-source pool, PCCL reaches 43.83, exceeding random at 43.05 and the strongest retained baseline, DPP, at 43.08. This smaller but consistent aggregate gain is accompanied by improvements across several heterogeneous sources: PCCL is strongest on SQuAD, SelfInst, S-NI, and UnNI, while remaining competitive on SamSum and GSM8K. The same pattern appears in the task-level results: PCCL improves Qwen over random on SQuAD (72.59 versus 70.83) and StrategyQA (51.72 versus 50.75), while GSM8K is essentially unchanged (45.48 versus 45.52). For the Llama replication, PCCL reaches 67.17 on SQuAD and 44.04 on GSM8K, compared with 65.48 and 43.36 for random.

Taken together, these results highlight the benefit of treating curriculum design as a joint problem of when training signals are introduced and which signals are combined within each update. PCCL explicitly addresses both dimensions through principal-loading-based ordering and cross-source batch construction, yielding more effective supervision than approaches that organize training along only one axis.

4.3 Ablation Study

Table 4 isolates the contributions of PCCL's two components in the multi-source setting. Cross-source batch construction alone improves several benchmarks over random batching, while principal-loading ordering provides complementary gains by prioritizing examples that better reflect the dominant structure of the training pool. Combining both components yields the strongest overall performance, indicating that batch composition and learning order play distinct but complementary roles in determining which signals are learned together and when they are emphasized.



(a) Convergence over optimizer steps

(b) Sensitivity to batch size

Figure 3: (a) ROUGE-L over optimizer steps for PCCL and representative baselines in multi-source distillation. PCCL reaches strong performance earlier in training. (b) Sensitivity of PCCL to batch size, showing final ROUGE-L across batch sizes from 16 to 256.

4.4 Faster Convergence

Figure 3(a) tracks pooled ROUGE-L against optimizer steps for PCCL and representative curriculum baselines on the multi-source setting. PCCL rises earlier and remains above random, GraB, and CLPD through most of training, reaching approximately 43.5 by step 300 and 43.8 by step 350. Random reaches a similar score only at the final checkpoint, while GraB and CLPD remain lower. The curves therefore support a convergence-speed advantage for PCCL in this protocol: it obtains a strong fraction of its final score with fewer optimizer steps.

4.5 Sensitivity Analysis

Figure 3(b) evaluates the sensitivity of PCCL to batch size over a broad range from 16 to 256. ROUGE-L increases from 42.56 at batch size 16 to 42.91 at 32 and 43.83 at 64, after which the performance remains stable at 43.64 and 43.86 for batch sizes 128 and 256, respectively. The improvement from small to moderate batch sizes is expected, as larger batches allow each update to incorporate a broader mixture of training signals. More importantly, performance does not degrade as the batch size continues to increase. These results indicate that PCCL is not dependent on a narrowly tuned batch configuration. Its principal-loading order remains effective across different optimization granularities, while the cross-source construction continues to preserve heterogeneous supervision within each batch. This robustness suggests that the benefit of PCCL comes from how training signals are organized, rather than from a particular choice of batch size.

5 Conclusion

Curriculum design for knowledge distillation involves more than deciding which examples should be introduced first. In heterogeneous training pools, it must also determine which learning signals should be combined within the same update. We introduced PCCL to address these two dimensions jointly. PCCL derives a curriculum order from the principal structure of per-example teacher gradients, prioritizing examples that capture more shared information in the training pool, and complements this ordering with cross-source batch construction to preserve diversity within each update. Across both single-source and multi-source distillation, PCCL consistently improves student performance over random batching and competitive curriculum baselines, while also reaching strong performance earlier in training. The gains are reproduced across Qwen and Llama students and are further supported by ablations showing that both ordering and batch composition contribute to the final performance. These results suggest that effective distillation curricula should move beyond scalar notions of difficulty and instead organize supervision along both when training signals are introduced and which signals are learned together.

AI USE STATEMENT

In this work, we used generative AI tools solely for manuscript editing to improve language and clarity, and for assistance with code development. We did not use generative AI tools for any other tasks. We reviewed all AI-assisted edits to ensure that they accurately preserved our intended meaning and verified and tested AI-assisted code for correctness. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Knowledge Distillation can transfer a teacher's biases and errors to the student. The models and datasets remain subject to their licenses, and attribution agreement does not by itself establish safety or factual reliability.

REFERENCES

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

Prayas Agrawal, Prateek Chanda, Ishita Khatri, Ganesh Ramakrishnan, Bamdev Mishra, and Pratik Jawanpuria. Minibatch selection for language models via partition matroid constrained gradient matching. *arXiv preprint arXiv:2606.07954*, 2026.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Jincheng Cao, Fanzhi Zeng, Leqi Liu, and Aryan Mokhtari. Curriculum learning-guided progressive distillation in large language models. *arXiv preprint arXiv:2605.11260*, 2026.

Sahil Chaudhary. Code alpaca: An instruction-following llama model for code generation. `https://github.com/sahil280114/codealpaca`, 2023.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`.

Jaewoong Cho, Kartik Sreenivasan, Keon Lee, Kyunghoo Mun, Soheun Yi, Jeong-Gwan Lee, Anna Lee, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. Mini-batch optimization of contrastive loss. *arXiv preprint arXiv:2307.05906*, 2023.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

Talfan Evans, Nikhil Parthasarathy, Hamza Merzić, and Olivier J Henaff. Data curation via joint example selection further accelerates multimodal learning. *Advances in Neural Information Processing Systems*, 37:141240–141260, 2024.

Fartash Faghri, David Duvenaud, David J Fleet, and Jimmy Ba. A study of gradient variance in deep learning. *arXiv preprint arXiv:2007.04532*, 2020.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *CoRR*, abs/1911.12237, 2019. URL `http://arxiv.org/abs/1911.12237`.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8ac015d409635f196f9e3e9dcfb9a94e-Abstract-Conference.html.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14409–14428, 2023.

Yaning Jia, Chunhui Zhang, Xingjian Diao, Xiangchi Yuan, Zhongyu Ouyang, Chiyu Ma, and Soroush Vosoughi. What makes a good curriculum? disentangling the effects of data ordering on llm mathematical reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 34472–34488, 2026.

Zishang Jiang, Jinyi Han, Tingyun Li, Xinyi Wang, Sihang Jiang, Xiaojun Meng, Jiansheng Wei, Jiaqing Liang, and Yanghua Xiao. Difficulty is not enough: Curriculum learning for LLMs fine-tuning must consider utility. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. doi: 10.1609/aaai.v40i37.40400. URL https://ojs.aaai.org/index.php/AAAI/article/view/40400.

Krishnateja Killamsetty, Durga S, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. GRAD-MATCH: Gradient matching based data subset selection for efficient deep model training. In *ICML*, 2021. URL https://proceedings.mlr.press/v139/killamsetty21a.html.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, 2016. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

M Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23, 2010.

Kasper Green Larsen, Rasmus Pagh, and Jakub Tetek. Countsketches, feature hashing and the median of three. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6011–6020. PMLR, 2021. URL http://proceedings.mlr.press/v139/larsen21a.html.

Bruce W Lee, Hyunsoo Cho, and Kang Min Yoo. Instruction tuning with human curriculum. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 1281–1309, 2024. doi: 10.18653/v1/2024.findings-naacl.82. URL https://aclanthology.org/2024.findings-naacl.82/.

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

Zheng Li, Xiang Li, Lingfeng Yang, Borui Zhao, Renjie Song, Lei Luo, Jun Li, and Jian Yang. Curriculum temperature for knowledge distillation. In Brian Williams, Yiling Chen, and Jennifer Neville (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 1504–1512. AAAI Press, 2023. doi: 10.1609/AAAI.V37I2.25236. URL https://doi.org/10.1609/aaai.v37i2.25236.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. Regmix: Data mixture as regression for language model pre-training. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=5BjQOUXq7i.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982.

Yucheng Lu, Wentao Guo, and Christopher De Sa. Grab: Finding provably better data permutations than random reshuffling. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/3acb49252187efa352a1ae0e4b066ced-Abstract-Conference.html.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6950–6960, 2020. URL https://proceedings.mlr.press/v119/mirzasoleiman20a.html.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK: Attributing model behavior at scale. In *ICML*, 2023. URL https://proceedings.mlr.press/v202/park23c.html.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3962–3971. IEEE, 2019.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. Competence-based curriculum learning for neural machine translation. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 1162–1172, 2019.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, 2020.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In Jian Su, Xavier Carreras, and Kevin Duh (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 2383–2392. The Association for Computational Linguistics, 2016. doi: 10.18653/V1/D16-1264. URL https://doi.org/10.18653/v1/d16-1264.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van

Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

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 *NeurIPS*, 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ed165f2ff227cf36c7e3ef88957dadd9-Paper-Conference.pdf.

Peiqi Wang, Yikang Shen, Zhen Guo, Matthew Stallone, Yoon Kim, Polina Golland, and Rameswar Panda. Diversity measurement and subset selection for instruction tuning datasets. *arXiv preprint arXiv:2402.02318*, 2024b. URL https://arxiv.org/abs/2402.02318.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pp. 5085–5109, 2022.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 13484–13508, 2023.

Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th International Conference on Machine Learning*, 2009. URL https://alex.smola.org/papers/2009/Weinbergeretal09.pdf.

Daphna Weinshall, Gad Cohen, and Dan Amir. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *International conference on machine learning*, pp. 5238–5246. PMLR, 2018.

Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. When do curricula work? *arXiv preprint arXiv:2012.03107*, 2020.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning (ICML)*, 2024.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. In Alice Oh, Tristan Naumann, Amir

Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/dcba6be91359358c2355cd920da3fcbd-Abstract-Conference.html.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yuanhao Yue, Chengyu Wang, Jun Huang, and Peng Wang. Distilling instruction-following abilities of large language models with task-aware curriculum planning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 6030–6054, 2024.

Cheng Zhang, Hedvig Kjellström, and Stephan Mandt. Determinantal point processes for mini-batch diversification. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, 2017. URL https://auai.org/uai2017/proceedings/papers/69.pdf.

Jipeng Zhang, Yaxuan Qin, Renjie Pi, Weizhong Zhang, Rui Pan, and Tong Zhang. Tagcos: Task-agnostic gradient clustered coreset selection for instruction tuning data. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4686–4701, 2025.

Peilin Zhao and Tong Zhang. Accelerating minibatch stochastic gradient descent using stratified sampling. *arXiv preprint arXiv:1405.3080*, 2014. URL https://arxiv.org/pdf/1405.3080.

Yang Zhao, Li Du, Xiao Ding, Yangou Ouyang, Hepeng Wang, Kai Xiong, Jinglong Gao, Zhouhao Sun, Dongliang Xu, Qing Yang, Dongchen Li, Bing Qin, and Ting Liu. Beyond similarity: A gradient-based graph method for instruction tuning data selection. In *ACL*, 2025. URL https://aclanthology.org/2025.acl-long.1189/.

A Details of Experimental Settings

A.1 Datasets

- Dolly-15k (Conover et al., 2023): a human-written instruction corpus containing question-answering, information-extraction, and summarization examples. We sample $4{,}000$ examples for the single-source training pool.
- SQuAD (Rajpurkar et al., 2016): a reading-comprehension dataset in which answers are extracted from provided passages. We use $1{,}000$ examples in the multi-source training pool and report token-level F1 on its held-out evaluation examples.
- SAMSum (Gliwa et al., 2019): a dialogue-summarization dataset pairing conversational transcripts with concise summaries. We use $1{,}000$ training examples and include its held-out set in the multi-source in-distribution evaluation.
- CodeAlpaca (Chaudhary, 2023): an instruction-following corpus focused on code-generation and code-understanding tasks. We use $1{,}000$ examples to provide a programming-oriented source in the multi-source pool.
- GSM8K (Cobbe et al., 2021): a collection of grade-school mathematical word problems requiring multi-step reasoning. We use $1{,}000$ training examples and report answer accuracy on its held-out evaluation examples.
- SelfInst (Wang et al., 2023): an instruction-following evaluation set used to measure transfer beyond the training sources.
- Vicuna (Chiang et al., 2023): a multi-turn instruction-following evaluation set derived from user-oriented conversations and used here as an out-of-distribution benchmark.
- Super-NaturalInstructions (Wang et al., 2022): a broad collection of natural-language tasks expressed through declarative instructions. We use its held-out examples as an out-of-distribution evaluation source.
- Unnatural Instructions (Honovich et al., 2023): a synthetically expanded instruction-following benchmark designed to test generalization to diverse instruction forms.

A.2 Baselines

In this section, we provide a detailed description of the baseline methods used for comparison.

- random: This control globally reshuffles the training pool before each epoch.
- DPP (Zhang et al., 2017): The authors use a determinantal point process to favor diverse subsets with large determinant values under a similarity kernel. The resulting mini-batches discourage redundant examples while preserving the fixed batch size.
- FCL (Bengio et al., 2009): The authors introduce curriculum learning by presenting examples from easier to harder according to a chosen difficulty measure. The schedule is intended to let the learner acquire simpler patterns before more difficult ones.
- RCL (Bengio et al., 2009): This comparator reverses the difficulty order and presents harder examples first. It tests whether the direction of a difficulty curriculum matters.
- GFC (Jia et al., 2026): The authors divide examples into three difficulty tiers and visit the tiers from easy to hard. Shuffling within each tier separates tier-level scheduling from a deterministic example order.
- GRC (Jia et al., 2026): This variant visits the same three difficulty tiers from hard to easy. It tests the reverse direction while keeping the tier construction fixed.
- CLPD (Cao et al., 2026): The authors use learning progress as a signal for deciding which data should receive training exposure. Their curriculum couples progress measurement with a changing teacher schedule.
- IFD-order (Li et al., 2024): The authors score examples by instruction-following difficulty to identify data that are more or less useful for instruction tuning. The score is used here as an ordering signal rather than as a subset-selection rule.

Table 5: Multi-source distillation with Llama-3.1-8B → Llama-3.2-1B. Scores are ROUGE-L (×100, mean ± SD over five seeds); All is the pooled mean over evaluation examples.

Method	SQuAD	SamSum	GSM8K	SelfInst	Vicuna	S-NI	UnNI	All
Composition baselines								
DPP	61.9 ±3.0	38.7 ±0.6	**36.8 ±0.7**	15.9 ±0.8	14.0 ±0.6	28.9 ±1.9	32.8 ±1.7	37.05 ±0.87
Ordering baselines								
FCL	63.0 ±0.5	33.4 ±0.4	35.0 ±1.1	14.1 ±0.5	11.0 ±0.7	25.6 ±1.4	29.9 ±1.3	34.63 ±0.51
RCL	52.5 ±1.7	34.2 ±1.9	22.1 ±5.0	13.1 ±1.3	6.2 ±1.2	26.8 ±1.9	31.3 ±2.1	30.86 ±1.64
GFC	64.4 ±1.0	38.7 ±0.5	35.2 ±0.8	15.2 ±0.4	**14.2 ±0.6**	28.5 ±2.4	32.3 ±2.1	37.01 ±0.65
GRC	60.4 ±2.3	37.1 ±0.7	35.3 ±0.9	14.7 ±0.8	10.5 ±1.1	27.4 ±2.5	33.2 ±2.5	35.82 ±0.91
CLPD	62.2 ±2.6	34.4 ±1.3	34.5 ±0.2	14.1 ±0.4	11.6 ±0.6	24.3 ±0.5	32.7 ±1.1	34.90 ±0.57
IFD-order	55.3 ±2.7	35.4 ±1.0	23.2 ±2.3	14.7 ±0.8	9.1 ±1.0	**33.0 ±0.8**	**35.4 ±1.7**	33.81 ±1.32
GraB	65.5 ±2.2	38.6 ±0.6	35.8 ±0.4	15.3 ±0.6	13.6 ±1.1	27.3 ±1.1	28.8 ±2.3	36.42 ±0.44
CORGI	64.0 ±1.2	**39.0 ±0.5**	36.3 ±0.4	14.8 ±0.9	13.7 ±0.4	28.1 ±1.1	32.7 ±1.8	37.12 ±0.55
DUCL	64.1 ±1.2	36.0 ±0.3	35.4 ±0.4	13.7 ±0.8	11.7 ±0.5	26.2 ±1.6	30.6 ±1.1	35.57 ±0.50
Loss schedule								
CTKD	61.8 ±1.1	37.1 ±1.3	34.4 ±0.5	15.3 ±0.8	13.2 ±1.0	29.0 ±1.3	31.4 ±3.3	36.03 ±0.97
random	65.1 ±0.8	38.3 ±0.6	35.6 ±1.2	15.8 ±0.9	13.3 ±1.3	29.9 ±2.6	33.8 ±2.3	37.66 ±0.59
PCCL	**66.8 ±1.1**	38.9 ±0.8	36.0 ±0.3	**16.7 ±1.4**	13.3 ±0.6	32.0 ±1.8	33.4 ±0.7	**38.51 ±0.58**

- GraB (Lu et al., 2022): The authors construct an example permutation by balancing cumulative gradients through a herding procedure. The goal is to make prefixes of the order approximate the gradient contribution of the full dataset.
- CORGI (Lee et al., 2024): The authors interleave examples across groups within successive difficulty levels. This combines difficulty-based progression with diversity of group membership.
- DUCL (Jiang et al., 2026): The authors gradually expand a window of eligible examples as training proceeds. The window begins with easier data and eventually exposes the full pool.
- CTKD (Li et al., 2023): The authors vary the distillation temperature over training to change the softness of the teacher target. This is a loss-schedule curriculum rather than a data-ordering rule.

A.3 Teacher Adaptation

For every teacher model used in our experiments, we perform one LoRA-based supervised adaptation on the same $4{,}000$-example pool used for student distillation. After this adaptation, the teacher is frozen throughout all student-training runs. We use LoRA rank 16, alpha 32, and dropout 0, and apply the adapters to the query, key, value, output, gate, up-projection, and down-projection layers in every Transformer block. For the Qwen3-8B teacher, this makes 43.6 million of the model's 8.23 billion parameters trainable, corresponding to 0.53%; all remaining teacher parameters stay fixed during distillation.

B Additional Experimental Results

Table 6 and 5 report the Llama replication of both the single-source and multi-source distillation settings. We follow the same training protocol and evaluation setup as in the main Qwen experiments, replacing the model family while keeping the curriculum construction unchanged. This replication is intended to examine whether the gains of PCCL generalize beyond a single teacher–student architecture.

Table 6: Single-source distillation with Llama-3.1-8B → Llama-3.2-1B. Scores are ROUGE-L (×100, mean ± SD over five seeds); All is the pooled mean over evaluation examples.

Method	DollyEval	SelfInst	Vicuna	S-NI	UnNI	All
Composition baselines						
DPP	23.7 ±0.7	15.7 ±0.7	15.9 ±0.7	24.3 ±1.0	26.5 ±0.7	23.25 ±0.11
Ordering baselines						
FCL	23.3 ±0.3	16.4 ±0.3	14.4 ±0.3	24.3 ±0.3	25.6 ±0.4	22.90 ±0.13
RCL	22.6 ±0.8	15.9 ±0.7	11.6 ±0.8	25.3 ±2.6	**28.1 ±2.2**	23.49 ±1.39
GFC	23.4 ±0.2	16.5 ±0.1	16.7 ±0.1	25.5 ±0.5	26.2 ±0.3	23.54 ±0.18
GRC	23.4 ±0.3	15.2 ±0.7	14.8 ±0.5	25.4 ±1.0	27.5 ±1.1	23.61 ±0.56
CLPD	22.4 ±0.3	16.2 ±0.4	13.7 ±0.4	23.9 ±0.5	25.1 ±0.6	22.34 ±0.14
IFD-order	23.8 ±0.8	15.3 ±0.7	11.5 ±1.0	24.0 ±1.0	26.2 ±1.5	22.81 ±0.72
GraB	22.3 ±0.7	15.3 ±0.2	16.6 ±0.4	23.2 ±0.5	25.0 ±0.9	22.12 ±0.39
CORGI	22.7 ±0.3	16.3 ±0.4	15.8 ±0.5	**25.9 ±0.3**	26.9 ±0.4	23.57 ±0.21
DUCL	23.4 ±0.3	15.4 ±0.3	16.0 ±0.3	25.8 ±0.3	26.4 ±0.3	23.48 ±0.23
Loss schedule						
CTKD	22.7 ±0.6	15.3 ±0.6	15.5 ±1.3	22.9 ±1.0	24.9 ±0.8	22.08 ±0.41
random	23.7 ±0.5	15.9 ±0.9	**16.8 ±0.6**	23.7 ±1.4	25.9 ±0.4	22.96 ±0.58
PCCL	**23.9 ±0.8**	**16.7 ±0.7**	15.7 ±0.8	25.5 ±1.2	27.2 ±1.6	**23.90 ±0.91**