CONVEX MULTI-TARGET DATASET SELECTION FOR POST-TRAINING VIA PORTFOLIO OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training *multiple* target capabilities under a fixed auxiliary-data budget is harder than single-target selection because one auxiliary mixture must account for transfer across potentially conflicting targets. Inspired by single-target data selection with Kernel Mean Matching (KMM), we view this problem through the lens of Markowitz portfolio optimization: auxiliary datasets are assets, target-auxiliary alignment is return, and the similarity matrix among auxiliary datasets is the risk in the capital asset pricing model. This view naturally extends to multiple targets by letting a user-specified preference vector define the aggregate return across target capabilities, while the redundancy penalty among auxiliary datasets remains shared. Based on this formulation, we introduce Multi-Target KMM (MT-KMM), which uses linear scalarization to compute continuous dataset scores, then post-processes them into a common auxiliary training mixture for multiple targets. Motivated by Adam's first update from zero moments, we represent each dataset by its normalized signed gradient at a frozen reference model, which coincides with Adam's normalized first-step features from zero moments in the zero-stabilizer limit. The gradients are computed only once, and the resulting selection can be reused to train models up to 70B. Theoretically, we show that finite-preview MT-KMM valuations converge to their population counterparts at the $\widetilde{O}(m^{-1/2})$ rate for raw gradients and, under a linear near-zero condition, for SignGD, where m is the size of the preview gradients. Empirically, MT-KMM achieves the highest mean target accuracy among the compared methods at every tested Llama scale and auxiliary-data budget for SFT, with gains extending to Qwen models from 0.6B to 32B and to RL post-training. By varying the target preference vector across different target capabilities, we show that MT-KMM can adapt the selected auxiliary mixture to different downstream priorities.

1 INTRODUCTION

Post-training often augments limited target data with a larger pool of auxiliary data (Thulke et al., 2024; Colombo et al., 2024; Singhal et al., 2025). Existing methods study which examples or source tasks to select for fine-tuning (Xia et al., 2024; Li et al., 2024; Dai et al., 2025) and how to set their mixture proportions (Renduchintala et al., 2024; Li et al., 2025). We assume target data are already included in post-training; the question is how to spend a fixed auxiliary training budget. With multiple target tasks, one shared auxiliary mixture should support several capabilities according to their relative priorities. This setting is increasingly relevant in modern LLM post-training: Qwen3 reports both positive and negative transfer across targeted capabilities after broad RL (Yang et al., 2025), while Llama 3 repeatedly rebalances its post-training mixture against a broad evaluation suite chosen of interest (Grattafiori et al., 2024).

To approach the above problem, we start from the single-target Kernel Mean Matching (KMM) formulation of Zeng et al. (2026). Its objective has the structure of mean-variance portfolio optimization (Markowitz, 1952): auxiliary datasets act as assets, target alignment acts as return, and the quadratic gradient-similarity term plays the role of portfolio risk, analogous to the covariance penalty on correlated assets. Thus, KMM rewards useful target transfer while discouraging redundant auxiliary allocation. The multi-target problem is then how to define portfolio return when several target capabilities matter and potentially conflict with each other.

We develop Multi-Target KMM (MT-KMM) through linear scalarization as shown in Fig. 1. A user-specified preference vector α weights target-specific alignment signals, while the quadratic auxiliary interaction term is shared across any type of preferences. Unlike Zeng et al. (2026), we represent auxiliary and target datasets by normalized signed gradients (SignGD) at a frozen reference model.

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.

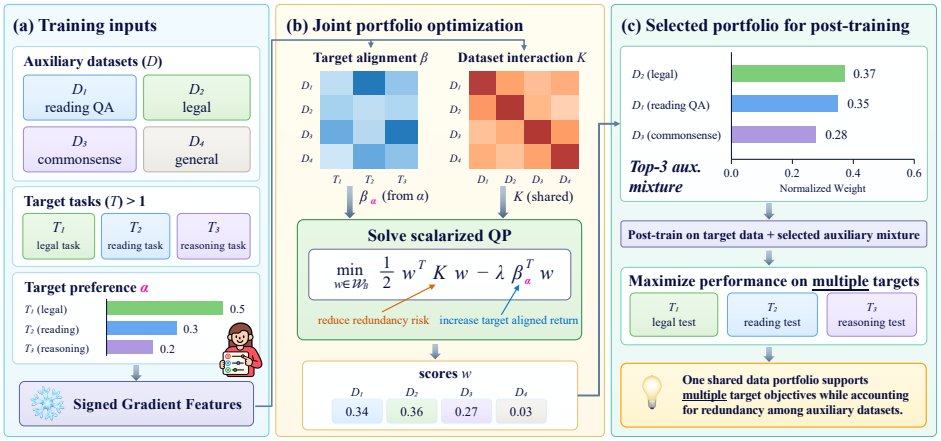

Figure 1: MT-KMM overview. SignGD represents auxiliary datasets and target tasks. A pre-defined user preference vector α scalarizes target priorities; MT-KMM balances the resulting multi-objective target alignment against auxiliary redundancy to obtain continuous scores, then retains the top-k positive scores and normalizes them for post-training.

For any α, a convex quadratic program first computes continuous dataset scores, which are then post-processed into the auxiliary training mixture.

Our main contributions are: **(i)** a preference-conditioned multi-target KMM formulation that reuses a single auxiliary kernel across target preferences, allowing the selected data mixture to adapt efficiently if downstream priorities evolve; **(ii)** grounded justifications for the use of SignGD as dataset representations, together with a theoretical characterization of the statistical stability of the resulting continuous scores; and **(iii)** empirical evidence that selections computed on a small frozen model transfer across model scales, families, and post-training algorithms.

2 Related Work

Multi-target data mixture optimization. Several recent methods optimize training mixtures with respect to multiple downstream targets. Pretraining-focused approaches include TiKMiX (Wang et al., 2025) and GRAPE (Fan et al., 2026), which dynamically optimize source-domain mixtures across multiple validation targets, while CAMEL (Li et al., 2026) and OP-Mix (Hu et al., 2026) further support user-specified weighted downstream objectives. Multi-target mixture optimization has also been studied for post-training, including AutoMixAlign (Corrado et al., 2025) for DPO and MOSAIC (Dou & Yang, 2026) for iterative multi-objective supervised fine-tuning mixture search. DynaMiCS (Gualdoni et al., 2026) considers multi-domain fine-tuning by partitioning evaluation domains into target domains to improve and constrained domains whose losses should remain below specified reference levels, dynamically optimizing mixture weights subject to these preservation constraints. Applying this constrained formulation to our preference-specified setting would additionally require converting relative target preferences into target/constraint assignments and absolute constraint thresholds, a mapping that is generally nontrivial.

Multi-target instance-level data selection. A separate line of work considers selecting *individual training examples or documents* for multiple downstream targets under a data budget. BIDS (Dai et al., 2025) targets balanced learning across multiple capabilities by selecting the most influential individual example for the currently underrepresented task. ParetoHqD (Gu et al., 2026) performs preference-aware instance selection for multi-objective alignment by selecting examples according to their position in a multi-dimensional reward space, while NAG (Wang et al., 2026) and G2IS (Zhao et al., 2025) consider document- or example-level selection with respect to multiple target tasks. These methods therefore operate at the instance level, whereas our decision unit is an entire auxiliary dataset and the auxiliary training budget constrains the amount of data drawn from selected datasets. Influence- or gradient-based instance-level selection can additionally face scalability and estimation-stability challenges when large numbers of individual examples must be scored (Pezeshkpour et al., 2021; Basu et al., 2020). Our formulation instead assigns preference-dependent values directly to candidate datasets. Adjacent single- and multi-objective literatures are in Sec. A.

3 PRELIMINARIES

Setup. Let $\theta_0 \in \mathbb{R}^d$ denote the parameters of a pretrained model. We are given N candidate auxiliary datasets $\mathcal{D}_{\text{aux}} = \{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$. Each dataset $\mathcal{D}_i$ is associated with a population distribution P_i and risk $L_i(\theta) := \mathbb{E}_{x\sim P_i}[\ell(x;\theta)]$, where $\ell(x;\theta)$ denotes the per-example post-training loss. We write $g_i := \nabla_\theta L_i(\theta_0)$ for the population gradient induced by auxiliary dataset $\mathcal{D}_i$. We additionally consider a target distribution P_{tar}, with population risk $L_{\text{tar}}(\theta) := \mathbb{E}_{x\sim P_{\text{tar}}}[\ell(x;\theta)]$ and target gradient $g_{\text{tar}} := \nabla_\theta L_{\text{tar}}(\theta_0)$. A dataset valuation method learns a vector of continuous dataset valuation scores $w = (w_1, \ldots, w_N) \in \mathbb{R}^N$, where w_i represents the relative importance assigned to auxiliary dataset $\mathcal{D}_i$. The learned scores are post-processed to construct a training mixture under a separately specified auxiliary training budget.

3.1 DATASET SELECTION AS PORTFOLIO CONSTRUCTION

Selecting auxiliary datasets can be cast as a resource allocation problem, where a limited auxiliary training budget must be distributed across multiple candidate data sources. This setting shares the same structural principle as portfolio construction in finance.

Consider N candidate assets, where each asset i has a random return variable R_i. Each asset is characterized by its expected return $\mathbb{E}[R_i]$ and risk measured by its variance $\text{Var}(R_i)$. An investor can either allocate all capital to a single asset or construct a **portfolio**, *i.e.,* a joint combination of assets, by assigning weights w to each candidate, producing the *aggregate return* $R_w = \sum_{i=1}^N w_i R_i$. The expected return and risk of a portfolio are therefore $\mathbb{E}[R_w] = w^\top \mu$ and $\text{Var}(R_w) = w^\top \Sigma w$, where $\mu_i = \mathbb{E}[R_i]$ and $\Sigma_{ij} = \text{Cov}(R_i, R_j)$. The classic mean-variance formulation of Markowitz (1952) shows that a rational investor selects portfolio weights by balancing these two quantities:

$$\max_w \quad \mathbb{E}[R_w] - \frac{\gamma}{2}\text{Var}(R_w) = \max_w \quad w^\top \mu - \frac{\gamma}{2} w^\top \Sigma w \qquad \text{s.t.} \quad \mathbf{1}^\top w = 1,\ w \geq 0. \tag{1}$$

where γ controls the weighting between the intent to maximize return vs. minimize risk.

A well-constructed portfolio can achieve a better return-risk trade-off than selecting individual assets in isolation. For example, suppose two assets satisfy $\mathbb{E}[R_1] = \mathbb{E}[R_2] = 0.1$ and $\text{Var}(R_1) = \text{Var}(R_2) = 0.04$. An equally weighted portfolio with $w = (0.5, 0.5)$ preserves the expected return of 0.1, while achieving lower variance, $\text{Var}(R_w) = 0.02$, when the two assets are uncorrelated.

3.2 SINGLE-TARGET KERNEL MEAN MATCHING

The same allocation principle arises in auxiliary data selection for a single target task. Collect the auxiliary gradients into $\Phi := [g_1, \ldots, g_N] \in \mathbb{R}^{d\times N}$, such that a valuation vector w defines the *aggregate update direction* Φw. We evaluate the local utility of this score vector by the target-loss reduction after one local update of learning rate η,

$$u(w) := L_{\text{tar}}(\theta_0) - L_{\text{tar}}(\theta_0 - \eta\Phi w).$$

A second-order Taylor expansion around θ_0 gives

$$u(w) = \eta\langle g_{\text{tar}}, \Phi w\rangle - \frac{\eta^2}{2}(\Phi w)^\top H_{\text{tar}}(\Phi w) + O\left(\eta^3 \|\Phi w\|^3\right),$$

where $H_{\text{tar}} := \nabla^2_\theta L_{\text{tar}}(\theta_0)$. Under the locally isotropic approximation $H_{\text{tar}} \approx cI$ for some $c > 0$,

$$u(w) \approx \eta\beta^\top w - \frac{c\eta^2}{2} w^\top K w, \qquad K := \Phi^\top \Phi, \quad \beta := \Phi^\top g_{\text{tar}}.$$

Here, $\beta_i = \langle g_i, g_{\text{tar}}\rangle$ measures the first-order local target improvement associated with auxiliary dataset $\mathcal{D}_i$. Under the isotropic approximation, the objective has exactly the same quadratic form as the Markowitz mean–variance objective in Eq. (1), with $\mu := \eta\beta$ and $\Sigma := c\eta^2 K \succeq 0$ for the equivalent portfolio problem. Thus, $K_{ij} = \langle g_i, g_j\rangle$, up to scaling, determines the covariance coupling between auxiliary datasets: assigning scores of the same sign to datasets with similar update directions increases the portfolio risk. Motivated by this equivalence, Zeng et al. (2026) considers the feasible set $\mathcal{W}_B := \{w \in \mathbb{R}^N : \|w\|_1 \leq B\}$ with $B \geq 0$ the ℓ_1 score-radius parameter, and solves a convex quadratic program

$$w^\star \in \arg\min_{w\in\mathcal{W}_B} \ \frac{1}{2} w^\top K w - \lambda\beta^\top w, \tag{2}$$

where $\lambda > 0$ controls the relative importance of target alignment. Completing the square yields the equivalent least-squares formulation

$$w^\star \in \arg\min_{w \in \mathcal{W}_B} \frac{1}{2} \|\Phi w - \lambda g_{\mathrm{tar}}\|_2^2 . \tag{3}$$

This formulation can be interpreted as kernel mean matching (KMM; Gretton et al., 2009), where the gradient inner product induces a neural-tangent-style kernel over datasets (Jacot et al., 2018). The resulting objective encourages the weighted auxiliary update direction to match the target update direction while accounting for redundancy among auxiliary datasets.

4 Multi-Target KMM with Signed-Gradient Representations

4.1 Multi-Target Kernel Mean Matching (MT-KMM)

We now extend the portfolio perspective to multiple downstream targets. Consider M target objectives indexed by $r \in [M]$, where target r is associated with a distribution $P_{\mathrm{tar},r}$, population risk $L_{\mathrm{tar},r}(\theta) := \mathbb{E}_{x \sim P_{\mathrm{tar},r}}[\ell_r(x;\theta)]$, and gradient $g_{\mathrm{tar},r} := \nabla_\theta L_{\mathrm{tar},r}(\theta_0)$. A downstream application may assign different importance to these objectives. We represent these preferences by $\alpha \in \Delta_M := \{\alpha \in \mathbb{R}_+^M : \sum_{r=1}^M \alpha_r = 1\}$ and define the preference-weighted target loss

$$L_{\mathrm{tar},\alpha}(\theta) := \sum_{r=1}^M \alpha_r L_{\mathrm{tar},r}(\theta), \qquad g_{\mathrm{tar},\alpha} := \nabla_\theta L_{\mathrm{tar},\alpha}(\theta_0) = \sum_{r=1}^M \alpha_r g_{\mathrm{tar},r}.$$

As in the single-target setting, a valuation vector w defines the aggregate auxiliary direction Φw. Let $\beta_r := \Phi^\top g_{\mathrm{tar},r}$ be per-target alignment vector. The local isotropic approximation to $L_{\mathrm{tar},\alpha}$ gives

$$L_{\mathrm{tar},\alpha}(\theta_0) - L_{\mathrm{tar},\alpha}(\theta_0 - \eta\Phi w) \approx \eta \beta_\alpha^\top w - \frac{c\eta^2}{2} w^\top K w, \qquad \beta_\alpha := \Phi^\top g_{\mathrm{tar},\alpha} = \sum_{r=1}^M \alpha_r \beta_r.$$

Therefore, for a given target preference α, the corresponding continuous score vector solves

$$w_\alpha^\star \in \arg\min_{w \in \mathcal{W}_B} \frac{1}{2} w^\top K w - \lambda \beta_\alpha^\top w. \tag{4}$$

The auxiliary interaction term K is shared across target preferences, while α changes the target-alignment return through β_α. Thus, once auxiliary dataset interactions have been characterized, different downstream preferences can be accommodated by changing only the linear term of the same quadratic program. Equivalently, associate each target with the single-target matching objective $F_r(w) := 1/2 \cdot \|\Phi w - \lambda g_{\mathrm{tar},r}\|_2^2$. Then minimizing Eq. (4) is equivalent, up to an additive constant independent of w, to minimizing the linearly scalarized objective

$$F_\alpha(w) := \sum_{r=1}^M \alpha_r F_r(w) = \frac{1}{2} \|\Phi w - \lambda g_{\mathrm{tar},\alpha}\|_2^2 + \mathrm{const}. \tag{5}$$

This gives MT-KMM both a preference-weighted downstream-utility interpretation and a multi-objective distribution-matching interpretation. Since each F_r and the feasible set $\mathcal{W}_B$ are convex, F_α remains a convex optimization problem. Moreover, for strictly positive α, every minimizer of F_α is Pareto optimal with respect to $(F_1, \ldots, F_M)$; varying α therefore traces preference-dependent Pareto-optimal score vectors for these continuous matching objectives.

From dataset scores to a training mixture. For our chosen feasible set $\mathcal{W}_B$, it is possible to obtain negative valuation scores. We discard nonpositive scores and retain up to k datasets with the largest positive scores, where k is the dataset-count limit. Denote their indices by $\mathrm{TopK}_+(w)$. For a nonempty retained set, we normalize the scores to sum to one, i.e., $\widetilde{w}_i = w_i / \sum_{j \in \mathrm{TopK}_+(w)} w_j$ for $i \in \mathrm{TopK}_+(w)$ and $\widetilde{w}_i = 0$ otherwise. We then sample auxiliary datasets according to $\widetilde{w}$, allocating the auxiliary training budget in these proportions. The score radius B, dataset-count limit k, and auxiliary training budget are separate hyperparameters that depend on the downstream task.

4.2 Signed Gradient as Dataset Representatives

The MT-KMM formulation above is agnostic to the choice of representation Φ. Following Zeng et al. (2026), one can use raw gradients, which encode both directional and magnitude information.

However, gradient magnitudes can vary systematically with task-dependent loss scales (Chen et al., 2018) and sequence length in instruction tuning (Xia et al., 2024), potentially biasing valuation toward datasets for reasons unrelated to transfer. We therefore consider signed gradients (SignGD) as an alternative representation that only preserves coordinate-wise directional information, which admits two additional properties relevant in large-scale model training.

Local Optimization Geometry of SignGD. Let $g := \nabla_\theta L(\theta_0)$. A first-order expansion of loss with updated model weight gives $L(\theta_0 + \Delta) = L(\theta_0) + \langle g, \Delta \rangle + O(\|\Delta\|_2^2)$. Restricting the maximum coordinate-wise change of Δ to $\eta > 0$ imposes a separate bound $|\Delta_j| \le \eta$ on each coordinate, so the linearized descent problem decomposes as

$$\min_{\|\Delta\|_\infty \le \eta} \langle g, \Delta \rangle = \sum_{j=1}^{d} \min_{|\Delta_j| \le \eta} g_j \Delta_j.$$

The above optimization problem is decomposable so that each coordinate can be optimized independently. If $g_j > 0$, decreasing Δ_j lowers $g_j \Delta_j$, so its minimum occurs at $\Delta_j = -\eta$; if $g_j < 0$, the minimum occurs at $\Delta_j = +\eta$. When $g_j = 0$, every feasible choice is optimal, and we take $\Delta_j = 0$. Thus, we have $\Delta^\star = -\eta \operatorname{sign}(g)$. The fast gradient sign method uses the same ℓ_∞ steepest-descent geometry to generate adversarial examples (Goodfellow et al., 2015), while sign-based updates have also been studied directly as optimization algorithms (Bernstein et al., 2018).

Motivation from Adam's first update. SignGD is particularly natural at our local selection point because Adam (Kingma & Ba, 2015) and variants such as AdamW (Loshchilov & Hutter, 2019) are standard optimizers for modern LLM training (Malladi et al., 2023). With zero-initialized moments $m_0 = v_0 = 0$ and decay rates $\beta_1, \beta_2 \in [0, 1)$, the first Adam step gives $m_1 = (1 - \beta_1)g$ and $v_1 = (1 - \beta_2)g^2$; after bias correction, $\widehat{m}_1 = g$ and $\widehat{v}_1 = g^2$. Substituting these estimates into the parameter update yields

$$\Delta_{1,j}^{\mathrm{Adam}} = -\eta \frac{\widehat{m}_{1,j}}{\sqrt{\widehat{v}_{1,j}} + \epsilon_{\mathrm{Adam}}} = -\eta \frac{g_j}{|g_j| + \epsilon_{\mathrm{Adam}}} \approx -\eta \operatorname{sign}(g_j), \tag{6}$$

where the approximation holds when $|g_j| \gg \epsilon_{\mathrm{Adam}}$. Motivated by this local update representation, we instantiate

$$\phi_i := \frac{1}{\sqrt{d}} \operatorname{sign}(g_i), \qquad \phi_{\mathrm{tar},r} := \frac{1}{\sqrt{d}} \operatorname{sign}(g_{\mathrm{tar},r}). \tag{7}$$

These features induce the symmetric SignGD kernel $\kappa_{\mathrm{SignGD}}(g, g') := d^{-1} \langle \operatorname{sign}(g), \operatorname{sign}(g') \rangle$, with $\|\phi\|_2 \le 1$. Thus, K_{ij} and $(\beta_r)_i$ measure auxiliary redundancy and target alignment in a common signed geometry, and Eq. (4) performs kernel mean matching in this feature space. We emphasize that this signed-feature objective is a matching surrogate rather than a direct target-loss influence interpretation. In contrast, LESS (Xia et al., 2024) evaluates SignGD within the asymmetric influence term $\langle \nabla_\theta \ell(z_{\mathrm{val}}; \theta_t), \Gamma_{\mathrm{Adam}}(z; \theta_t) \rangle$, retaining the raw validation gradient; MT-KMM instead applies SignGD symmetrically to both auxiliary and target distributions.

4.3 Finite-Sample Valuation Analysis

Data selection often takes place before access to the full datasets, so dataset representations are estimated from small amount of finite sampled gradients, which we call *previews*. Suppose that each auxiliary and target representation is estimated from a preview of m i.i.d. samples. Let $\widehat{\Phi} := [\widehat{\phi}_1, \ldots, \widehat{\phi}_N]$ denote the empirical auxiliary representation matrix, and for each $r \in [M]$, let $\widehat{\phi}_{\mathrm{tar},r}$ denote the empirical target representation. For $\alpha \in \Delta_M$, define $\widehat{\phi}_{\mathrm{tar},\alpha} := \sum_{r=1}^{M} \alpha_r \widehat{\phi}_{\mathrm{tar},r}$, and $\widehat{\beta}_\alpha := \widehat{\Phi}^\top \widehat{\phi}_{\mathrm{tar},\alpha}$. Let $w_\alpha^\star$ denote the population continuous score vector and $\widehat{w}_\alpha$ denote its empirical counterpart computed from finite auxiliary and target samples:

$$w_\alpha^\star := \arg\min_{w \in \mathcal{W}_B} \frac{1}{2} w^\top K w - \lambda \beta_\alpha^\top w, \qquad \widehat{w}_\alpha := \arg\min_{w \in \mathcal{W}_B} \frac{1}{2} w^\top \widehat{K} w - \lambda \widehat{\beta}_\alpha^\top w.$$

Without loss of generality, for the following finite-sample results, we set $\lambda = 1$.

Theorem 4.1 (Finite-sample stability of raw gradient MT-KMM scores)**.** *Assume that the sample-level representations are bounded in ℓ_2 norm by a universal constant C and $\lambda_{\min}(K) \ge \mu > 0$.*

Then, for sufficiently large m, w.p. at least $1-\delta$, simultaneously for every $\alpha \in \Delta_M$,

$$\|\widehat{w}_\alpha - w^\star_\alpha\|_2 \le \widetilde{O}\left(\frac{(B+1)\sqrt{N}}{\mu\sqrt{m}}\right), \tag{8}$$

where $\widetilde{O}(\cdot)$ hides logarithmic factors in $N, M, d, 1/\delta$, and C is treated as a constant.

Remark 4.1. With m preview samples per dataset, the MT-KMM score bound accounts for both auxiliary- and target-estimation uncertainty while retaining the $\widetilde{O}(m^{-1/2})$ rate. Since target preferences form convex combinations with $\alpha \in \Delta_M$, target-estimation errors do not accumulate across objectives. See full proof in Sec. B.1.

SignGD takes signs after averaging gradients, so its finite-sample analysis must account for sign instability near zero. For $t \ge 0$, let $p(t)$ denote the largest fraction of population-gradient coordinates with absolute value at most t across auxiliary and target datasets. The function $p(t) \in [0,1]$ is nondecreasing in t. These coordinates lie close to 0 that estimation error may change their signs.

Theorem 4.2 (Finite-sample stability of SignGD MT-KMM scores)**.** *Under the same assumptions as Thm. 4.1, w.p. at least $1-\delta$, simultaneously for all $\alpha \in \Delta_M$,*

$$\|\widehat{w}_\alpha - w^\star_\alpha\|_2 \le \frac{4(B+1)\sqrt{N}}{\mu}\, p(\tau), \qquad \tau = \widetilde{O}(m^{-1/2}). \tag{9}$$

In particular, if $p(t)$ near zero increases at most linearly with the neighborhood width t, then $\|\widehat{w}_\alpha - w^\star_\alpha\|_2 = \widetilde{O}(m^{-1/2})$ uniformly over α.

Remark 4.2. Thm. 4.2 is an asymptotic, margin-dependent stability result. As the preview size m increases, the uncertainty threshold $\tau = \widetilde{O}(m^{-1/2})$ shrinks. The bound uses a worst-case coordinate concentration scale and is not intended as a tight numerical certificate at the finite preview sizes used in our experiments. Two corollaries connect this score-stability guarantee to Adam initialized with zero moments in the zero-stabilizer limit. See full proof in Sec. B.2.

- (Cor. B.7) SignGD's population KMM solution coincides with that induced by Adam's normalized first-step features in the zero-ϵ limit, so Eq. (9) controls deviation from this reference without a population mismatch term.
- (Cor. B.8): For raw-gradient KMM, the bound includes a representation-mismatch term that need not vanish, even with exact population gradients.

5 Experiments

5.1 Experiment Setup

Benchmark and candidate pool. We evaluate the eight targets in Tab. 1, all with finite answer sets. Our pool contains N=96 auxiliary datasets, with 50 sharing a task family with at least one target, and 46 additional ones deliberately broaden the candidate set. This mix tests whether a selector can distinguish directly related and redundant sources from a bigger pool of candidates that may or may not transfer. Sec. C.1 include additional dataset details.

(a) Target benchmark

Target	Task Description	# Labels
LEDGAR	legal classification	100
RACE	exam reading comp.	4
QuAIL	reading comp. / QA	4
ReClor	passage-based logic	4
LogiQA2.0	passage-based logic	4
CommonsenseQA	commonsense QA	5
ANLI	entailment / neutral / contradiction	3
Banking77	banking intent	77

(b) Auxiliary candidate pool

Target-related (50). Reading comprehension/QA (12), commonsense/science MC (13), textual entailment (7), logical reasoning (4), intent/topic classification (6), and legal (8).
E.g., SQuAD, HellaSwag, SNLI, ProofWriter, CLINC, BillSum.
Broad contrast / distractor (46). Sentiment/text classification (5), mathematics (14), code (8), medical (6), finance (3), and general instruction (10).
E.g., SST-2, MetaMath, MBPP, WikiDoc, Financial PhraseBank, UltraChat.

Table 1: Benchmark and candidate pool. All targets have finite answer sets and are scored by accuracy. Pool counts are gathered by task family; shaded rows give representative examples. "Distractor" means no direct family match with target tasks.

(a) Llama SFT

Method	1B	8B	70B
target only	31.10	54.40	60.50
random	32.57	54.77	63.56
antiselect	31.57	53.47	63.16
GradEx-RE	36.39	59.53	64.84
GradEx-FS	35.55	56.98	63.27
DataModel	33.47	57.77	64.90
DataModel-CS	33.21	55.85	64.35
One Step	39.51	59.23	67.79
MT-KMM	**43.42**	**61.63**	**69.38**

(b) Qwen3 SFT

Method	0.6B	8B	32B
target only	40.40	64.40	66.40
random	43.55	66.57	67.84
antiselect	43.70	66.34	68.44
One Step	45.81	68.74	68.99
MT-KMM	**46.60**	**69.38**	**69.39**

(c) Llama GRPO

Method	Accuracy
target only	37.90
random	38.44
antiselect	37.60
One Step	40.82
MT-KMM	**42.33**

Table 2: Mean held-out accuracy (%; 5 seeds). **(a,b)** SFT on 8 targets; **(c)** GRPO on 4 targets in Sec. D.3 (Llama-3.2-1B). Gray rows give the target-only reference numbers. The best result per column is **bold**.

Data selection configuration. By default, we use all 8 targets, the 96-dataset pool, signGD features, the uniform target composite $\alpha_r{=}1/M$, and use a dataset-count limit of $k{=}10$. We estimate each target and auxiliary gradient from 256 gradient samples on frozen pretrained checkpoints. The reference grid covers all combinations of $t \in \{20, 40, 100, 200\}$ target training examples per task and total auxiliary training budgets of 3200 or 6400 examples, giving $4 \times 2 = 8$ combinations. To reduce computation, Llama-3.1-70B uses five settings: it omits $t{=}40$ and evaluates $t{=}20$ only with 3200 auxiliary examples. Selection is run once on the smallest model in each family, Llama-3.2-1B (Grattafiori et al., 2024) or Qwen3-0.6B (Yang et al., 2025), and the postprocessed training mixture is reused for larger models (Tab. 22). Solver and post-processing details are in Sec. B.3.

Training protocol and evaluation metrics. By default, we use 1 epoch of LoRA (Hu et al., 2021) supervised fine-tuning (SFT) with results on 5 seeds; more training hardware and hyperparameter details are in Sec. C.3. Evaluation uses a held-out target test slice of up to 800 examples per target unseen during gradient extraction, selection, or training. For each method, we first average accuracy over seeds, and then report the unweighted mean across the eight targets and the target-size and auxiliary-data settings evaluated for each model.

5.2 Results

MT-KMM performs best across training paradigms and model scales. Across the benchmark described in Sec. 5.1, Tab. 2 shows that MT-KMM has the highest mean held-out accuracy at every tested Llama and Qwen model scale under SFT and remains best under GRPO. Following Zeng et al. (2026), we adapt the baselines used in the single-target setting. The SFT comparison includes One Step gradient alignment, variants of GradEx (Li et al., 2024) and DataModel (Ilyas et al., 2022), random selection, and target-only training. We also include antiselect as a negative control, which chooses the auxiliary datasets with the lowest MT-KMM scores. The GradEx and DataModel implementations used here rely on the SFT next-token-loss setting, so the GRPO comparison excludes them. We give implementation details, explain our multi-target modifications in Sec. C.2, and GRPO setups in Sec. D.3. MT-KMM is also best at every target count and under both auxiliary-budget settings in Fig. 2a, although its gain over target-only training decreases as more tasks are optimized jointly. The narrowing gap between budget settings reflects their converging total auxiliary budgets, while the remaining decline may reflect the increasing difficulty of finding a single auxiliary portfolio that benefits a larger and more diverse target set.

MT-KMM benefits most from auxiliary data when target supervision is limited. As target supervision increases, the advantage of auxiliary data narrows: at 2000 examples per target, MT-KMM exceeds target-only by only 1.8 points (Fig. 2b). In contrast, with only 200 target examples per task, scaling auxiliary data improves MT-KMM by 11.0 points (Fig. 2c).

The selected datasets empirically stabilize with larger previews. Fig. 3a complements the finite-preview guarantees in Sec. 4.3: larger previews make the induced top-k selection increasingly stable, even though the theory is stated for the underlying continuous scores. Downstream accuracy, however, changes little with preview size, in contrast to the 9.3–11.0 point gains obtained by scaling target or auxiliary training data in Fig. 2b–c. This suggests that preview-estimation noise is not the main bottleneck once the selection is reasonably stable. Rather, SignGD remains a local, one-step signal whose more precise estimation need not better predict the full fine-tuning trajectory. Task vec-

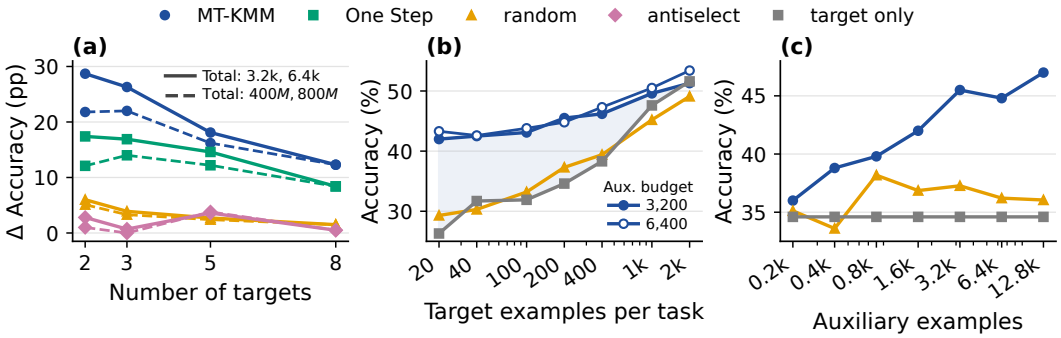

Figure 2: Scaling sweeps on Llama-3.2-1B. We define $\Delta\text{Acc} := \text{Acc}_{\text{method}} - \text{Acc}_{\text{target only}}$, in percentage points. **(a)** ΔAcc for nested sets of 2, 3, 5, and 8 targets. Solid lines average over fixed total auxiliary budgets $\{3200, 6400\}$; dashed lines average over budgets that scale with the number of targets M, taking values in $\{400M, 800M\}$ examples. The two budget schedules coincide at $M = 8$. **(b–c)** Mean accuracy (%) over the eight-target set. **(b)** Target examples per task vary; filled and hollow MT-KMM circles denote auxiliary budgets 3200 and 6400, respectively. Random uses 3200 auxiliary examples. **(c)** The auxiliary budget varies with 200 target examples per task.

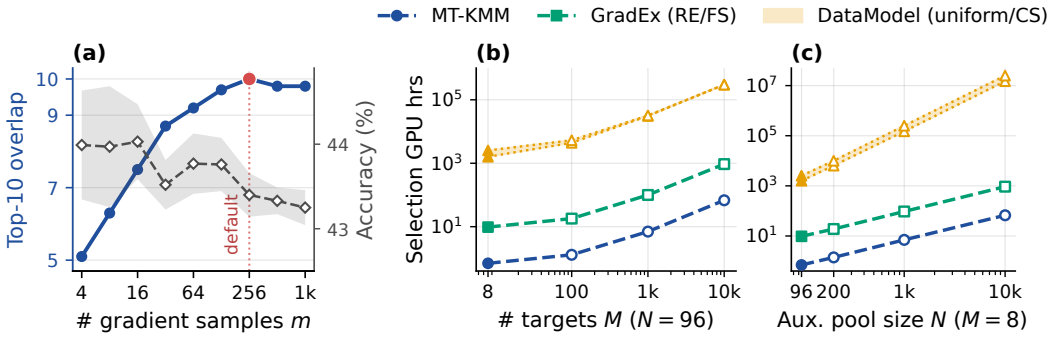

Figure 3: Selection stability and cost scaling. **(a)** Mean overlap between MT-KMM's top-10 selection at each m and its selection at the default m=256, over 10 draws; self-overlap at m=256 is 10 by definition. The dashed curve (right axis) shows downstream accuracy, with shading denoting 95% confidence intervals. **(b–c)** GPU-hours for selection methods as the target count M varies at N=96 and the auxiliary pool size N varies at M=8. Filled markers are measured empirically at our default setup; hollow markers are estimates obtained from the measured costs and each method's model-computation scaling rule. MT-KMM's projections cover gradient-pass GPU work; Gram-construction and QP costs are detailed in Sec. D.10. All GPU costs use Llama-3.2-1B on 8×H100.

tors (Ilharco et al., 2022), as used by Zeng et al. (2026), may better capture this trajectory through full fine-tuning updates.

MT-KMM keeps selection GPU work low as the number of targets and auxiliary datasets grows. Fig. 3b-c extrapolates GPU-hours for model computation during selection. MT-KMM's gradient passes and GradEx's fine-tuning scale as $O(N+M)$, with MT-KMM about $14\times$ cheaper at the reference setup. DataModel instead trains on many sampled auxiliary mixtures and its GPU work scales as $O(N(N+M))$. See Sec. D.10 for more details.

5.3 Analysis

Learned allocation strengthens preference steering. In Fig. 4a, increasing α, the weight on LEDGAR, shifts the auxiliary training budget from reading-comprehension data toward legal data. In Fig. 4b, selection with equal allocation gives only a weak response: as α increases from 0.3 to 0.7, LEDGAR accuracy rises by 1.0 percentage point and RACE accuracy falls by 0.2 points. Using the postprocessed mixture $\widetilde{w}$ for allocation produces larger changes of $+5.0$ and -1.2 points, respectively, and higher accuracy on both targets at every tested composite.

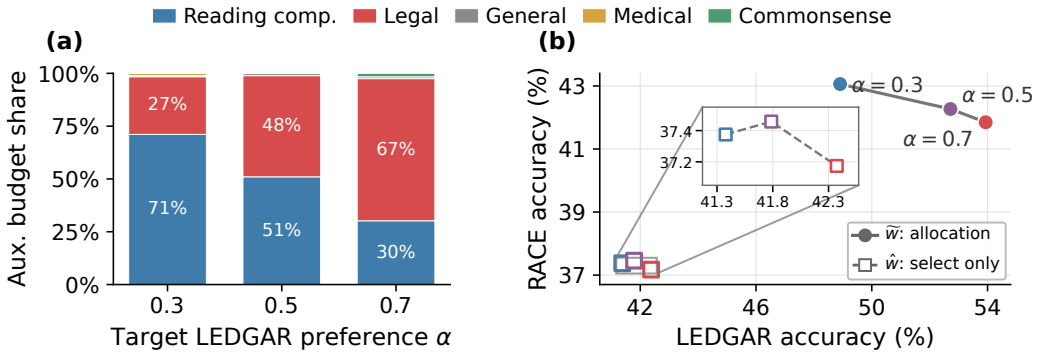

Figure 4: Preference steering for $M = 2$, with target composite $\phi_{\text{tar},\alpha} = \alpha\phi_{\text{LEDGAR}} + (1 - \alpha)\phi_{\text{RACE}}$ and k=5. **(a)** The auxiliary training budget shares among the top-5 selected auxiliaries, aggregated by auxiliary dataset type. Increasing α, the LEDGAR preference, shifts budget from reading-comprehension to legal data. **(b)** The resulting downstream performance trade-off. Each point is the trained model produced by one of the three target composites, with absolute LEDGAR accuracy (%) on the horizontal axis and absolute RACE accuracy (%) on the vertical axis. Solid circles use the postprocessed weights $\tilde{w}$ for allocation; hollow squares use $\hat{w}_\alpha$ to select datasets and split the auxiliary budget equally among them. Within each trajectory, α increases from left to right. Detailed dataset-level allocations and three-target results are in Sec. E.1.

(a) Comparison of gradient features

Targets	Raw	SignGD
LEDGAR + RACE	44.86	**47.49**
LEDGAR + ReClor	38.66	**39.66**
All 8 targets (Tab. 1)	42.62	**43.42**

(b) MT-KMM ablation

Method	1B	8B	70B
selection only	39.00	59.37	67.31
no K	40.52	60.26	67.80
MT-KMM	**43.42**	**61.63**	**69.38**

Table 3: Comparison of gradient features and MT-KMM key component ablations. **(a)** MT-KMM absolute accuracy (%) on Llama-3.2-1B. **Bold** marks the better feature in each row. **(b)** Absolute accuracy (%) on the Llama SFT setting as in Tab. 2a. "no K" selects the top-k positive target-alignment scores and allocates in proportion to them, without using the Gram matrix; "selection only" uses $\hat{w}_\alpha$ to select up to k datasets with the largest positive scores and splits the auxiliary budget equally among them, as defined in Sec. C.2.

SignGD improves over raw gradients. On Llama-3.2-1B (Tab. 3a), SignGD outperforms raw gradients on all three target sets, by 0.80–2.63 points. This is consistent with Sec. 4.2, since raw gradient magnitudes vary with loss scale and sequence length and can bias selection, while SignGD keeps only direction. We therefore use SignGD as the default feature.

Both auxiliary-dataset interactions and the learned allocation weights matter. In Tab. 3b, MT-KMM outperforms both ablations at every model scale. Replacing its full scoring and allocation procedure with the no-K control reduces accuracy by 1.37–2.90 points; equal allocation over MT-KMM's selected datasets costs 2.07–4.42 points. Under equal allocation, selection-only MT-KMM trails One Step by 0.51 and 0.48 points at 1B and 70B, respectively, and leads by just 0.14 points at 8B (Tab. 2a). Thus, selection alone does not consistently improve over One Step; learned allocation contributes materially to the full method's gains.

6 Conclusion

MT-KMM casts multi-target data selection as a gradient-space quadratic program that balances target alignment and inter-dataset redundancy while reusing a shared auxiliary Gram matrix across target preferences. Using gradients computed once on a small frozen model, MT-KMM consistently improves over baselines across model scales, families, data budgets, and fine-tuning algorithms. Ablations confirm the importance of both redundancy modeling and learned weighting.

Reproducibility statement

We describe the full experimental protocol in the main text and the appendix. We will release the source code upon acceptance.

Ethics statement

All datasets used in this work are publicly available and were obtained from their original public sources. We used this data only to compute aggregate statistics and to fine-tune models for measuring task accuracy. None of our evaluation tasks is medical, we perform no individual-level inference, and we make no clinical or medical claims.

AI use statement

Generative AI tools based on large language models were used as general-purpose assistants during the preparation of this paper. Their use was limited to writing and editing support, such as rephrasing, grammar, and LaTeX formatting, and to routine coding assistance for the plotting and table-generation scripts. All research ideas, the method, the experimental design, the analysis, and the scientific claims are the authors' own. The authors reviewed all AI-assisted text and code and take full responsibility for the content of the paper.

References

Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. Efficient online data mixing for language model pre-training. *arXiv preprint arXiv:2312.02406*, 2023.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. SmolLM2: when smol goes big – Data-Centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

b-mc2. sql-create-context dataset, 2023. URL `https://huggingface.co/datasets/b-mc2/sql-create-context`.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. TweetEval: unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1644–1650. Association for Computational Linguistics, 2020.

Samyadeep Basu, Philip Pope, and Soheil Feizi. Influence functions in deep learning are fragile. *arXiv preprint arXiv:2006.14651*, 2020.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on Freebase from Question-Answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1533–1544. Association for Computational Linguistics, 2013.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signSGD: Compressed optimisation for non-convex problems. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 560–569. PMLR, 2018. URL `https://proceedings.mlr.press/v80/bernstein18a.html`.

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439, 2020.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642. Association for Computational Linguistics, 2015.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, 2020.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Katz, and Nikolaos Aletras. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4310–4330, 2022.

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

Mayee Chen, Michael Hu, Nicholas Lourie, Kyunghyun Cho, and Christopher Ré. Aioli: A unified optimization framework for language model data mixing. In *International Conference on Learning Representations*, volume 2025, pp. 46089–46132, 2025.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. TheoremQA: a theorem-driven question answering dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7889–7901. Association for Computational Linguistics, 2023.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pp. 794–803. PMLR, 2018.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: exploring the surprising difficulty of natural Yes/No questions. 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. 2924–2936. Association for Computational Linguistics, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

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

Pierre Colombo, Telmo Pires, Malik Boudiaf, Rui Melo, Dominic Culver, Etienne Malaboeuf, Gabriel Hautreux, Johanne Charpentier, and Michael Desa. SaulLM-54B & SaulLM-141B: Scaling up domain adaptation for the legal domain. *Advances in Neural Information Processing Systems*, 37:129672–129695, 2024.

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`.

Nicholas E Corrado, Julian Katz-Samuels, Adithya M Devraj, Hyokun Yun, Chao Zhang, Yi Xu, Yi Pan, Bing Yin, and Trishul Chilimbi. AutoMixAlign: Adaptive data mixing for multi-task preference optimization in LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20234–20258, 2025.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*, 2018.

Ido Dagan, Oren Glickman, and Bernardo Magnini. *The PASCAL Recognising Textual Entailment Challenge*, pp. 177–190. Springer Berlin Heidelberg, 2006.

Qirun Dai, Dylan Zhang, Jiaqi W Ma, and Hao Peng. Improving influence-based instruction tuning data selection for balanced learning of diverse capabilities. In *EMNLP (Findings)*, pp. 7079–7102, 2025.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3029–3051. Association for Computational Linguistics, 2023.

Chandeepa Dissanayake, Lahiru Lowe, Sachith Gunasekara, and Yasiru Ratnayake. OpenBezoar: small, Cost-Effective and open models trained on mixes of instruction data. *arXiv preprint arXiv:2404.12195*, 2024.

Yipu Dou and Wang Yang. MOSAIC: Multi-objective slice-aware iterative curation for alignment. *arXiv preprint arXiv:2603.18637*, 2026.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: a reading comprehension benchmark requiring discrete reasoning over paragraphs. 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. 2368–2378. Association for Computational Linguistics, 2019.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. DoGE: Domain reweighting with generalization estimation. *arXiv preprint arXiv:2310.15393*, 2023.

Simin Fan, Maria Ios Glarou, and Martin Jaggi. GRAPE: Optimize data mixture for group robust multi-target adaptive pretraining. *Advances in Neural Information Processing Systems*, 38: 158023–158073, 2026.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did Aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL `https://arxiv.org/abs/1412.6572`.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, Bernhard Schölkopf, et al. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3 (4):5, 2009.

Haoran Gu, Handing Wang, Yi Mei, Mengjie Zhang, and Yaochu Jin. ParetoHqD: Fast offline multiobjective alignment of large language models using Pareto high-quality data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 17454–17462, 2026.

Eleonora Gualdoni, Sonia Laguna, Louis Bethune, Joao Monteiro, Pierre Ablin, and Marco Cuturi. DynaMiCS: Fine-tuning LLMs with performance constraints using dynamic mixtures. *arXiv preprint arXiv:2605.10770*, 2026.

Tianyu Han, Lisa C. Adams, Jens-Michalis Papaioannou, Paul Grundmann, Tom Oberhauser, Alexei Figueroa, Alexander Löser, Daniel Truhn, and Keno K. Bressem. MedAlpaca – an Open-Source collection of medical conversational AI models and training data. *arXiv preprint arXiv:2304.08247*, 2023.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. The ATIS spoken language systems pilot corpus. In *Proceedings of the workshop on Speech and Natural Language - HLT '90*, pp. 96–101. Association for Computational Linguistics, 1990.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Michael Y Hu, Apurva Gandhi, Kyunghyun Cho, Tal Linzen, and Pratyusha Sharma. Always learning, always mixing: Efficient and simple data mixing all the time. *arXiv preprint arXiv:2605.15220*, 2026.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data. In *International Conference on Machine Learning (ICML)*, 2022.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: a large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611. Association for Computational Linguistics, 2017.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 252–262. Association for Computational Linguistics, 2018.

Tushar Khot, Ashish Sabharwal, and Peter Clark. SciTaiL: a textual entailment dataset from science question answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018.

Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. GRAD-MATCH: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pp. 5464–5474. PMLR, 2021a.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. GLISTER: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 8110–8118, 2021b.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL `https://arxiv.org/abs/1412.6980`.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018.

Anastassia Kornilova and Vladimir Eidelman. BillSum: A corpus for automatic summarization of US legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 48–56, 2019.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model Post-Training. *arXiv preprint arXiv:2411.15124*, 2024.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. An evaluation dataset for intent classification and Out-of-Scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1311–1316. Association for Computational Linguistics, 2019.

Ariel N. Lee, Cole J. Hunter, and Nataniel Ruiz. Platypus: Quick, cheap, and powerful refinement of LLMs. *arXiv preprint arXiv:2308.07317*, 2023.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The Winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 2011.

Dongyue Li, Ziniu Zhang, Lu Wang, and Hongyang R Zhang. Scalable fine-tuning from multiple data sources: A first-order approximation approach. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 5608–5623, 2024.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems 36*, pp. 51991–52008. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023.

Jingwei Li, Xinran Gu, and Jingzhao Zhang. Capacity-aware mixture law enables efficient LLM data optimization. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=DDtL4VOUcT.

Yuan Li, Zhengzhong Liu, and Eric Xing. Data mixing optimization for supervised fine-tuning of large language models. *arXiv preprint arXiv:2508.11953*, 2025.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation with AlphaCode. *Science*, 378(6624):1092–1097, 2022.

Wing Lian, Guan Wang, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and Teknium. SlimOrca: an open dataset of GPT-4 augmented FLAN reasoning traces, with verification, 2023. URL https://huggingface.co/datasets/Open-Orca/SlimOrca.

Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. *Advances in neural information processing systems*, 32, 2019.

Xi Lin, Zhiyuan Yang, Qingfu Zhang, and Sam Kwong. Controllable Pareto multi-task learning. *arXiv preprint arXiv:2010.06313*, 2020.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 158–167. Association for Computational Linguistics, 2017.

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. WANLI: worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 6826–6847. Association for Computational Linguistics, 2022.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. LogiQA 2.0: An improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. RegMix: Data mixture as regression for language model pre-training. In *International Conference on Learning Representations*, volume 2025, pp. 38305–38339, 2025.

Zifan Liu, Amin Karbasi, and Theodoros Rekatsinas. TSDS: Data selection for task-specific model finetuning. *Advances in Neural Information Processing Systems*, 37:10117–10147, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150. Association for Computational Linguistics, 2011.

Debabrata Mahapatra and Vaibhav Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in Pareto optimization. In *International Conference on Machine Learning*, pp. 6597–6607. PMLR, 2020.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 23610–23641. PMLR, 2023. URL https://proceedings.mlr.press/v202/malladi23a.html.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796, 2014.

Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing English math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 975–984. Association for Computational Linguistics, 2020.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391. Association for Computational Linguistics, 2018.

Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. Orca-Math: unlocking the potential of SLMs in grade school math. *arXiv preprint arXiv:2402.14830*, 2024.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK: Attributing model behavior at scale. *arXiv preprint arXiv:2303.14186*, 2023.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094. Association for Computational Linguistics, 2021.

Pouya Pezeshkpour, Sarthak Jain, Byron C Wallace, and Sameer Singh. An empirical comparison of instance attribution methods for NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 967–975, 2021.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pp. 1–16, 2020.

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789. Association for Computational Linguistics, 2018.

Abhilasha Ravichander, Alan W Black, Shomir Wilson, Thomas Norton, and Norman Sadeh. Question answering for privacy policies: Combining computational and legal perspectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4946–4957. Association for Computational Linguistics, 2019.

Siva Reddy, Danqi Chen, and Christopher D Manning. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019.

HSVNS Kowndinya Renduchintala, Sumit Bhatia, and Ganesh Ramakrishnan. SMART: Submodular data mixture strategy for instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12916–12934, 2024.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 2011.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. Getting closer to AI complete question answering: A set of prerequisite real tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: an adversarial Winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740, 2020.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4462–4472. Association for Computational Linguistics, 2019.

Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of Chain-of-Thought. *arXiv preprint arXiv:2210.01240*, 2022.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. CARER: contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3687–3697. Association for Computational Linguistics, 2018.

Max Savery, Asma Ben Abacha, Soumya Gayen, and Dina Demner-Fushman. Question-driven summarization of answers to consumer health questions. *Scientific Data*, 7(1), 2020.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Haebin Shin, Lei Ji, Xiao Liu, Zhiwei Yu, Hyunwoo Yoo, Qi Chen, and Yeyun Gong. DynamixSFT: Dynamic mixture optimization of instruction tuning collections. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 39590–39603, 2026.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, et al. Toward expert-level medical question answering with large language models. *Nature medicine*, 31(3):943–950, 2025.

Ankur Sinha and Tanmay Khandait. *Impact of News on the Commodity Market: Dataset and Results*, pp. 589–601. Springer International Publishing, 2021.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642. Association for Computational Linguistics, 2013.

Haoran Sun, Lixin Liu, Junjie Li, Fengyu Wang, Baohua Dong, Ran Lin, and Ruohui Huang. Conifer: Improving complex constrained Instruction-Following ability of large language models. *arXiv preprint arXiv:2404.02823*, 2024.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. QuaRTz: an Open-Domain dataset of qualitative relationship questions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5940–5945. Association for Computational Linguistics, 2019.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. ProofWriter: generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3621–3634. Association for Computational Linguistics, 2021.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

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

Teknium. OpenHermes 2.5: An open dataset of synthetic data for generalist LLM assistants, 2023. URL `https://huggingface.co/datasets/teknium/OpenHermes-2.5`.

David Thulke, Yingbo Gao, Petrus Pelser, Rein Brune, Rricha Jalota, Floris Fok, Michael Ramos, Ian Van Wyk, Abdallah Nasir, Hayden Goldstein, et al. ClimateGPT: Towards AI synthesizing interdisciplinary research on climate change. *arXiv preprint arXiv:2401.09646*, 2024.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. OpenMathInstruct-2: accelerating AI for math with massive Open-Source instruction data. *arXiv preprint arXiv:2410.01560*, 2024.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. NewsQA: a machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pp. 191–200. Association for Computational Linguistics, 2017.

Joel A Tropp. An introduction to matrix concentration inequalities. *Foundations and trends® in machine learning*, 8(1-2):1–230, 2015.

Don Tuggener, Pius von Däniken, Thomas Peetz, and Mark Cieliebak. LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, 2020.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers Reinforcement Learning, 2020. URL `https://github.com/huggingface/trl`.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: a Multi-Task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355. Association for Computational Linguistics, 2018.

Steven Wang, Antoine Scardigli, Leonard Tang, Wei Chen, Dmitry Levkin, Anya Chen, Spencer Ball, Thomas Woodside, Oliver Zhang, and Dan Hendrycks. MAUD: an Expert-Annotated legal NLP dataset for merger agreement understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 16369–16382. Association for Computational Linguistics, 2023.

Yifan Wang, Binbin Liu, Fengze Liu, Yuanfan Guo, Jiyao Deng, Xuecheng Wu, Weidong Zhou, Xiaohuan Zhou, and Taifeng Wang. TiKMiX: Take data influence into dynamic mixture for language model pre-training. *arXiv preprint arXiv:2508.17677*, 2025.

Zijun Wang, Haoqin Tu, Weidong Zhou, Yiyang Zhou, Xiaohuan Zhou, Bingni Zhang, Weiguo Feng, Taifeng Wang, Cihang Xie, and Fengze Liu. Target-oriented pretraining data selection via neuron-activated graph. *arXiv preprint arXiv:2604.15706*, 2026.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106. Association for Computational Linguistics, 2017.

Adina Williams, Nikita Nangia, and Samuel Bowman. A Broad-Coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122. Association for Computational Linguistics, 2018.

Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Weidi Xie, and Yanfeng Wang. PMC-LLaMA: toward building open-source language models for medicine. *Journal of the American Medical Informatics Association*, 31(9):1833–1843, 2024.

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 S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. DoReMi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818, 2023.

Wanyun Xie, Francesco Tonin, and Volkan Cevher. Chameleon: A flexible data-mixing framework for language model pretraining and finetuning. *arXiv preprint arXiv:2505.24844*, 2025.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *International Conference on Learning Representations*, volume 2024, pp. 30745–30766, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: a dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380. Association for Computational Linguistics, 2018.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. In *International Conference on Learning Representations*, volume 2025, pp. 82263–82287, 2025.

Wenpeng Yin, Dragomir Radev, and Caiming Xiong. DocNLI: a large-scale dataset for document-level natural language inference. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 4913–4922. Association for Computational Linguistics, 2021.

Bei Yu, Yingya Li, and Jun Wang. Detecting causal language use in science findings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4663–4673. Association for Computational Linguistics, 2019.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. MetaMath: bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020a.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. ReClor: A reading comprehension dataset requiring logical reasoning. In *International Conference on Learning Representations (ICLR)*, 2020b.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MAmmoTH: building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*, 2023.

Xiang Yue, Tuney Zheng, Ge Zhang, and Wenhu Chen. MAmmoTH2: scaling instructions from the web. In *Advances in Neural Information Processing Systems 37*, pp. 90629–90660. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: a Large-Scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 93–104. Association for Computational Linguistics, 2018.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800. Association for Computational Linguistics, 2019.

Siqi Zeng, Christopher Jung, Rui Li, Zhe Kang, Ming Li, Nima Noorshams, Zhigang Wang, Fuchun Peng, Han Zhao, and Xue Feng. Convex dataset valuation for post-training. In *Forty-third International Conference on Machine Learning*, 2026. URL `https://openreview.net/forum?id=oAAL0wZSMF`.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. ReCoRD: bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*, 2018.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *arXiv preprint arXiv:1509.01626*, 2015.

Yang Zhao, Li Du, Xiao Ding, Yangou Ouyang, Hepeng Wang, Kai Xiong, Jinglong Gao, Zhouhao Sun, Dongliang Xu, Qing Yang, et al. Beyond similarity: A gradient-based graph method for instruction tuning data selection. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24391–24404, 2025.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Hong Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. SWIFT: A scalable lightweight infrastructure for fine-tuning. *arXiv preprint arXiv:2408.05517*, 2024.

Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. When does pretraining help? assessing self-supervised learning for law and the CaseHOLD dataset. In *Proceedings of the 18th International Conference on Artificial Intelligence and Law (ICAIL)*. Association for Computing Machinery, 2021. doi: 10.1145/3462757.3466088.

Tong Zhu, Daize Dong, Xiaoye Qu, Jiacheng Ruan, Wenliang Chen, and Yu Cheng. Dynamic data mixing maximizes instruction tuning for mixture-of-experts. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1663–1677, 2025.

APPENDIX CONTENTS

A Additional Related Work **21**

B Finite-Sample Analysis for Multi-Target KMM **21**

B.1 MT-KMM Score Stability with Raw Gradients 21

B.2 MT-KMM Score Stability with SignGD 24

B.3 Relation between the theoretical and implemented objectives 27

C Experimental setup **27**

C.1 Targets and auxiliary datasets 27

C.2 Baseline details 30

C.3 Training and evaluation details 32

D Additional results **33**

D.1 Detailed baseline comparisons 33

D.2 Additional Qwen results 34

D.3 Reinforcement learning with GRPO 34

D.4 Number of targets 35

D.5 Target size and auxiliary budget 35

D.6 Controlling for preview labels 35

D.7 Controlling for same-target auxiliaries 36

D.8 Auxiliary pool size 36

D.9 Gradient sample size 36

D.10 Selection cost and scalability 37

E Additional analysis **38**

E.1 Target preference and auxiliary allocation 38

E.2 Gradient source and model transfer 39

E.3 Statistical significance 40

A Additional Related Work

Single-objective data selection and mixture optimization. A broad machine learning literature studies data selection to improve training efficiency and downstream generalization, using methods such as validation-based subset selection (Killamsetty et al., 2021b), gradient matching (Killamsetty et al., 2021a), and training-data attribution (Park et al., 2023). Recent work has increasingly focused on target-aware, sample-level data selection for language-model post-training. LESS selects instruction-tuning examples according to their gradient influence on a target task (Xia et al., 2024), while TSDS uses task-specific distribution matching for fine-tuning data selection (Liu et al., 2024). Our setup instead treats each dataset as the decision unit. This dataset-level subset-selection setting is closer to GradEx, which efficiently estimates the performance of candidate subsets of source tasks (Li et al., 2024). In particular, our work extends the single-target KMM formulation for dataset-level valuation (Zeng et al., 2026). Closely related work instead optimizes mixture proportions across training domains or datasets. Much of this literature was developed for language-model pretraining (Xie et al., 2023; Fan et al., 2023; Albalak et al., 2023; Liu et al., 2025; Ye et al., 2025; Chen et al., 2025), post-training (Renduchintala et al., 2024; Zhu et al., 2025; Li et al., 2025; Shin et al., 2026), or is stage-agnostic (Xie et al., 2025). These methods also primarily optimize one single aggregate generalization criterion, while our setting considers multiple downstream target objectives.

Multi-objective learning. Our formulation is also related to multi-objective approaches in multi-task learning. Multi-task learning has been cast as multi-objective optimization when task losses conflict (Sener & Koltun, 2018), with subsequent work studying Pareto-optimal solutions and controllable trade-offs among tasks (Lin et al., 2019; 2020; Mahapatra & Rajan, 2020). Other approaches adapt task losses or gradients during joint optimization, including GradNorm and gradient surgery (Chen et al., 2018; Yu et al., 2020a). These methods optimize model parameters to trade off performance among multiple tasks; our optimization variable is instead a continuous dataset valuation vector, postprocessed into a mixture for subsequent training.

B Finite-Sample Analysis for Multi-Target KMM

B.1 MT-KMM Score Stability with Raw Gradients

We now account for finite-sample uncertainty in **both** the auxiliary and target representations. For simplicity, suppose that each of the N auxiliary datasets and M target objectives is represented using a preview of m *i.i.d.* samples.

Let

$$\Phi := [\phi_1, \dots, \phi_N] \in \mathbb{R}^{d\times N}, \qquad \widehat{\Phi} := [\widehat{\phi}_1, \dots, \widehat{\phi}_N] \in \mathbb{R}^{d\times N}$$

denote the population and empirical auxiliary representation matrices. For each target $r \in [M]$, let $\phi_{\mathrm{tar},r}$ and $\widehat{\phi}_{\mathrm{tar},r}$ denote its population and empirical representations. For any $\alpha \in \Delta_M$, define

$$\phi_{\mathrm{tar},\alpha} := \sum_{r=1}^{M} \alpha_r \phi_{\mathrm{tar},r}, \qquad \widehat{\phi}_{\mathrm{tar},\alpha} := \sum_{r=1}^{M} \alpha_r \widehat{\phi}_{\mathrm{tar},r}.$$

Correspondingly, define $K := \Phi^\top\Phi, \widehat{K} := \widehat{\Phi}^\top\widehat{\Phi}$, and $\beta_\alpha := \Phi^\top\phi_{\mathrm{tar},\alpha}, \widehat{\beta}_\alpha := \widehat{\Phi}^\top\widehat{\phi}_{\mathrm{tar},\alpha}$. The population and empirical continuous score vectors are

$$w^\star_\alpha := \arg\min_{w\in\mathcal{W}_B} \frac{1}{2} w^\top K w - \lambda\beta_\alpha^\top w, \qquad \widehat{w}_\alpha := \arg\min_{w\in\mathcal{W}_B} \frac{1}{2} w^\top \widehat{K} w - \lambda\widehat{\beta}_\alpha^\top w,$$

where $\mathcal{W}_B := \{w \in \mathbb{R}^N : \|w\|_1 \le B\}$ and $B \ge 0$ is the ℓ_1 score radius. We borrow the same two assumptions used in Zeng et al. (2026) and with out loss of generality, we assume $\lambda = 1$.

Assumption B.1 (Bounded sample representations)**.** Assume that the sample-level representations used to estimate the auxiliary and target representations are bounded in ℓ_2 norm by a universal constant C. Consequently, $\|\phi_i\|_2, \|\widehat{\phi}_i\|_2, \|\phi_{\mathrm{tar},r}\|_2, \|\widehat{\phi}_{\mathrm{tar},r}\|_2 \le C$ for every $i \in [N]$ and $r \in [M]$.

Assumption B.2 (Gram matrix positive definiteness)**.** The population Gram matrix satisfies $\lambda_{\min}(K) \ge \mu > 0$.

Proof sketch. For each $\alpha \in \Delta_M$, Lemma C.3 of Zeng et al. (2026) gives a similar residual decomposition statement. Applying the Matrix Bernstein argument (Tropp, 2015) separately to each auxiliary and target representation, followed by union bounds, gives separate uniform errors $\varepsilon_{\mathrm{aux}}$ and $\varepsilon_{\mathrm{tar}}$, whose logarithmic factors reflect the numbers of auxiliary datasets and target objectives, respectively. Using $\|\widehat{w}_\alpha\|_1 \le B$ to control the Gram-matrix perturbation and substituting these bounds into the residual decomposition yields the final upper bound.

Lemma B.3 (Uniform concentration of auxiliary representations)**.** *Under Assumption B.1, with probability at least* $1-\delta/2$*,*

$$\max_{i\in[N]} \left\|\widehat{\phi}_i - \phi_i\right\|_2 \le \varepsilon_{\mathrm{aux}},$$

where

$$\varepsilon_{\mathrm{aux}} := \sqrt{\frac{8C^2 L_{\mathrm{aux}}}{m}} + \frac{4CL_{\mathrm{aux}}}{3m}, \qquad L_{\mathrm{aux}} := \log\frac{2N(d+1)}{\delta}.$$

Proof. For a fixed auxiliary dataset $i \in [N]$, apply the proof of Lemma C.5 in Zeng et al. (2026) to the $1 \times d$ representation error $\widehat{\phi}_i^\top - \phi_i^\top$. The same boundedness argument gives a summand norm bounded by $2C/m$, while the variance parameter for this single row is bounded by $v_i \le \frac{4C^2}{m}$. Applying Matrix Bernstein with failure probability $\delta/(2N)$ therefore gives the final upper bound with a union bound over $i \in [N]$.

Compared with Lemma C.5 of Zeng et al. (2026), there are two differences. First, Lemma C.5 concentrates the entire $N \times d$ representation-error matrix, whose worst-case variance parameter is bounded by $4NC^2/m$, whereas the single-row variance above is bounded by $4C^2/m$. Second, concentrating the N rows separately requires a union bound. With failure probability $\delta/2$ allocated to the auxiliary concentration event, the matrix-level bound of Lemma C.5 contains the logarithmic factor $\log(2(N+d)/\delta)$, whereas the row-wise bound contains $\log(2N(d+1)/\delta)$. Thus, the row-wise argument removes a factor of N from the leading variance term at the cost of an additional logarithmic dependence on N from the union bound.

□

Lemma B.4 (Uniform concentration of target representations)**.** *Under Assumption B.1, with probability at least* $1-\delta/2$*,*

$$\max_{r\in[M]} \left\|\widehat{\phi}_{\mathrm{tar},r} - \phi_{\mathrm{tar},r}\right\|_2 \le \varepsilon_{\mathrm{tar}},$$

where

$$\varepsilon_{\mathrm{tar}} := \sqrt{\frac{8C^2 L_{\mathrm{tar}}}{m}} + \frac{4CL_{\mathrm{tar}}}{3m}, \qquad L_{\mathrm{tar}} := \log\frac{2M(d+1)}{\delta}.$$

Consequently, on the same event, simultaneously for every $\alpha \in \Delta_M$*,*

$$\left\|\widehat{\phi}_{\mathrm{tar},\alpha} - \phi_{\mathrm{tar},\alpha}\right\|_2 \le \varepsilon_{\mathrm{tar}}.$$

Proof. The first statement follows identically to Lem. B.3, applying a union bound over the M target objectives. For the second statement, define $e_r := \widehat{\phi}_{\mathrm{tar},r} - \phi_{\mathrm{tar},r}$. For any $\alpha \in \Delta_M$,

$$\left\|\widehat{\phi}_{\mathrm{tar},\alpha} - \phi_{\mathrm{tar},\alpha}\right\|_2 = \left\|\sum_{r=1}^M \alpha_r e_r\right\|_2 \le \sum_{r=1}^M \alpha_r \|e_r\|_2 \le \varepsilon_{\mathrm{tar}} \sum_{r=1}^M \alpha_r = \varepsilon_{\mathrm{tar}}.$$

Because the same concentration event controls all $r \in [M]$, the result holds simultaneously for all $\alpha \in \Delta_M$. □

Theorem B.5 (Finite-sample score stability with finite target and auxiliary data)**.** *Under Assumption B.1 and Assumption B.2, with probability at least* $1-\delta$*, simultaneously for every* $\alpha \in \Delta_M$*,*

$$\|\widehat{w}_\alpha - w_\alpha^\star\|_2 \le \frac{C\sqrt{N}}{\mu}\left[(2B+1)\varepsilon_{\mathrm{aux}} + \varepsilon_{\mathrm{tar}}\right]. \tag{10}$$

Consequently, in the regime $m \gtrsim \max\{L_{\mathrm{aux}}, L_{\mathrm{tar}}\}$*, treating* C *as a constant,*

$$\|\widehat{w}_\alpha - w_\alpha^\star\|_2 \le O\left(\frac{\sqrt{N}}{\mu\sqrt{m}}\left[(B+1)\sqrt{L_{\mathrm{aux}}} + \sqrt{L_{\mathrm{tar}}}\right]\right). \tag{11}$$

Proof. For any fixed $\alpha \in \Delta_M$, the population and empirical scalarized objectives have the same constrained quadratic form as in Lemma C.3 of Zeng et al. (2026), with β replaced by β_α. Since the scalarization only changes the linear term, the residual-decomposition argument applies directly and gives

$$\|\widehat{w}_\alpha - w_\alpha^\star\|_2 \le \frac{1}{\mu}\left[\left\|(\widehat{K}-K)\widehat{w}_\alpha\right\|_2 + \left\|\widehat{\beta}_\alpha - \beta_\alpha\right\|_2\right]. \tag{12}$$

It therefore suffices to control the Gram-matrix error and the target-alignment error.

To bound the Gram-matrix error in the first term of Eq. (12), in the event in Lem. B.3, $\max_{i\in[N]} \|\widehat{\phi}_i - \phi_i\|_2 \le \varepsilon_{\mathrm{aux}}$. For every $i, j \in [N]$, by Assumption B.1,

$$\left|\widehat{K}_{ij} - K_{ij}\right| = \left|\left\langle\widehat{\phi}_i, \widehat{\phi}_j\right\rangle - \langle\phi_i, \phi_j\rangle\right| = \left|\left\langle\widehat{\phi}_i - \phi_i, \widehat{\phi}_j\right\rangle + \left\langle\phi_i, \widehat{\phi}_j - \phi_j\right\rangle\right| \le 2C\varepsilon_{\mathrm{aux}}.$$

Rather than converting this entrywise error into a spectral norm bound on the entire Gram matrix in Zeng et al. (2026), we directly control its action on the feasible valuation vector. For each $i \in [N]$,

$$\left|\left[(\widehat{K}-K)\widehat{w}_\alpha\right]_i\right| \le \sum_{j=1}^{N}\left|\widehat{K}_{ij} - K_{ij}\right||(\widehat{w}_\alpha)_j| \le 2C\varepsilon_{\mathrm{aux}}\|\widehat{w}_\alpha\|_1 \le 2CB\varepsilon_{\mathrm{aux}},$$

where the last inequality follows from $\widehat{w}_\alpha \in \mathcal{W}_B$. Taking the ℓ_2 norm across the N coordinates gives

$$\left\|(\widehat{K}-K)\widehat{w}_\alpha\right\|_2 \le 2CB\sqrt{N}\,\varepsilon_{\mathrm{aux}}. \tag{13}$$

To bound the target-alignment error in the second term of Eq. (12), for every $i \in [N]$,

$$\begin{aligned}(\widehat{\beta}_\alpha)_i - (\beta_\alpha)_i &= \left\langle\widehat{\phi}_i, \widehat{\phi}_{\mathrm{tar},\alpha}\right\rangle - \langle\phi_i, \phi_{\mathrm{tar},\alpha}\rangle \\ &= \left\langle\widehat{\phi}_i - \phi_i, \widehat{\phi}_{\mathrm{tar},\alpha}\right\rangle + \left\langle\phi_i, \widehat{\phi}_{\mathrm{tar},\alpha} - \phi_{\mathrm{tar},\alpha}\right\rangle.\end{aligned} \tag{14}$$

Since $\alpha \in \Delta_M$, $\left\|\widehat{\phi}_{\mathrm{tar},\alpha}\right\|_2 \le \sum_{r=1}^{M}\alpha_r\left\|\widehat{\phi}_{\mathrm{tar},r}\right\|_2 \le C$. Moreover, by Lem. B.4, $\|\widehat{\phi}_{\mathrm{tar},\alpha} - \phi_{\mathrm{tar},\alpha}\|_2 \le \varepsilon_{\mathrm{tar}}$. Therefore, plug in to Eq. (14),

$$\left|(\widehat{\beta}_\alpha)_i - (\beta_\alpha)_i\right| \le C\left(\varepsilon_{\mathrm{aux}} + \varepsilon_{\mathrm{tar}}\right).$$

Taking the ℓ_2 norm across the N coordinates gives

$$\left\|\widehat{\beta}_\alpha - \beta_\alpha\right\|_2 \le C\sqrt{N}\left(\varepsilon_{\mathrm{aux}} + \varepsilon_{\mathrm{tar}}\right). \tag{15}$$

Substituting Eqs. (13) and (15) into Eq. (12) gives

$$\begin{aligned}\|\widehat{w}_\alpha - w_\alpha^\star\|_2 &\le \frac{1}{\mu}\left[2CB\sqrt{N}\,\varepsilon_{\mathrm{aux}} + C\sqrt{N}\left(\varepsilon_{\mathrm{aux}} + \varepsilon_{\mathrm{tar}}\right)\right] \\ &= \frac{C\sqrt{N}}{\mu}\left[(2B+1)\varepsilon_{\mathrm{aux}} + \varepsilon_{\mathrm{tar}}\right],\end{aligned}$$

which proves Eq. (10). Finally, Lems. B.3 and B.4 imply

$$\varepsilon_{\mathrm{aux}} = O\left(C\sqrt{\frac{L_{\mathrm{aux}}}{m}} + C\frac{L_{\mathrm{aux}}}{m}\right), \qquad \varepsilon_{\mathrm{tar}} = O\left(C\sqrt{\frac{L_{\mathrm{tar}}}{m}} + C\frac{L_{\mathrm{tar}}}{m}\right).$$

In the stated sample-size regime, the L_{aux}/m and L_{tar}/m terms are dominated by the corresponding square-root terms. Treating C as a constant and using $2B + 1 \le 2(B + 1)$ gives the stated rate, which after suppressing logarithmic factors recovers Eq. (8). □

B.2 MT-KMM Score Stability with SignGD

Unlike the raw-gradient representation, the signed-gradient map used in Sec. 4.2 is discontinuous at zero. Consequently, even a small perturbation to a gradient coordinate may change its sign if the population coordinate lies sufficiently close to zero, whereas coordinates whose magnitudes exceed the estimation error cannot change sign. Therefore, the finite-sample stability of signed gradients is governed by the proportion of population-gradient coordinates lying near zero. This also means that the representation-concentration (Matrix Bernstein) step used in Thm. B.5 does not apply directly, since $\widehat{\phi}_i = d^{-1/2}\operatorname{sign}(\widehat{g}_i)$ is not an empirical average of sample-level representations. Instead, we first control the coordinate-wise gradient-estimation error and use the low-margin structure to bound the Gram matrix and target-alignment entries directly. The residual-decomposition argument from Thm. B.5 then gives a bound proportional to the fraction of low-margin coordinates. We make this dependence explicit below.

Theorem B.6 (Finite-sample stability of signed-gradient KMM scores)**.** *Assume that the per-example raw gradients satisfy Assumption B.1 and that the population signed-gradient Gram matrix satisfies Assumption B.2. For each auxiliary dataset $i \in [N]$, let*

$$\phi_i := \frac{1}{\sqrt{d}}\operatorname{sign}(g_i), \qquad \widehat{\phi}_i := \frac{1}{\sqrt{d}}\operatorname{sign}(\widehat{g}_i),$$

where $\widehat{g}_i := \frac{1}{m}\sum_{j=1}^{m}\nabla_\theta \ell(x_{ij};\theta_0)$ and $\operatorname{sign}(0) = 0$. Similarly, for each target objective $r \in [M]$, let

$$\phi_{\mathrm{tar},r} := \frac{1}{\sqrt{d}}\operatorname{sign}(g_{\mathrm{tar},r}), \qquad \widehat{\phi}_{\mathrm{tar},r} := \frac{1}{\sqrt{d}}\operatorname{sign}(\widehat{g}_{\mathrm{tar},r}),$$

where $\widehat{g}_{\mathrm{tar},r} := \frac{1}{m}\sum_{j=1}^{m}\nabla_\theta \ell_r(x_{rj};\theta_0)$. Define

$$\tau(\delta) := C\sqrt{\frac{2}{m}\log\frac{4(N+M)d}{\delta}}.$$

For $t \geq 0$, define the maximum fractions of low-margin coordinates among auxiliary and target gradients as

$$p_{\mathrm{aux}}(t) := \max_{i\in[N]}\frac{1}{d}\left|\{q \in [d] : |(g_i)_q| \leq t\}\right|, \quad p_{\mathrm{tar}}(t) := \max_{r\in[M]}\frac{1}{d}\left|\{q \in [d] : |(g_{\mathrm{tar},r})_q| \leq t\}\right|.$$

Write $p(t) := \max\{p_{\mathrm{aux}}(t), p_{\mathrm{tar}}(t)\}$, which is nondecreasing and takes values in $[0,1]$. With probability at least $1-\delta$, simultaneously for every $\alpha \in \Delta_M$,

$$\|\widehat{w}_\alpha - w^\star_\alpha\|_2 \leq \frac{4(B+1)\sqrt{N}}{\mu}p\big(\tau(\delta)\big). \tag{16}$$

If, for sufficiently small t, $p(t) \leq A_{\mathrm{mar}}t$ for some constant $A_{\mathrm{mar}} > 0$, then the bound is $\widetilde{O}(m^{-1/2})$.

Proof. We first control the coordinate-wise estimation errors of the auxiliary and target gradients. Under Assumption B.1, every gradient coordinate is bounded in absolute value by C. Therefore, for any auxiliary dataset $i \in [N]$ and coordinate $q \in [d]$, Hoeffding's inequality gives

$$\Pr\left(|(\widehat{g}_i)_q - (g_i)_q| > \tau\right) \leq 2\exp\left(-\frac{m\tau^2}{2C^2}\right).$$

A union bound over the Nd auxiliary dataset-coordinate pairs, with failure probability $\delta/2$ allocated to this event, yields

$$\max_{i\in[N]}\|\widehat{g}_i - g_i\|_\infty \leq \tau(\delta) \tag{17}$$

with probability at least $1-\delta/2$. Similarly, a union bound over the Md target-coordinate pairs gives

$$\max_{r\in[M]}\|\widehat{g}_{\mathrm{tar},r} - g_{\mathrm{tar},r}\|_\infty \leq \tau(\delta) \tag{18}$$

with probability at least $1-\delta/2$. Hence, Eqs. (17) and (18) hold simultaneously with probability at least $1-\delta$.

We now condition on this joint event. For any auxiliary dataset $i \in [N]$ and coordinate $q \in [d]$, suppose first that $(g_i)_q > \tau(\delta)$. On the event in Eq. (17),

$$|(\widehat{g}_i)_q - (g_i)_q| \le \tau(\delta),$$

and therefore

$$(\widehat{g}_i)_q \ge (g_i)_q - \tau(\delta) > 0.$$

Hence the empirical and population coordinates have the same positive sign. Similarly, if

$$(g_i)_q < -\tau(\delta),$$

then

$$(\widehat{g}_i)_q \le (g_i)_q + \tau(\delta) < 0,$$

so the two coordinates again have the same sign. Thus, any coordinate whose population magnitude exceeds $\tau(\delta)$ cannot change sign under the empirical estimation error. Equivalently, by contraposition,

$$\operatorname{sign}((\widehat{g}_i)_q) \ne \operatorname{sign}((g_i)_q) \quad \Longrightarrow \quad |(g_i)_q| \le \tau(\delta). \tag{19}$$

Therefore, the set of coordinates whose signs can differ is contained in the set of low-margin coordinates. We now bound the Gram matrix and target alignments directly. Each Gram entry is an average of sign products,

$$K_{ij} = \frac{1}{d}\sum_{q=1}^{d} \operatorname{sign}((g_i)_q)\operatorname{sign}((g_j)_q),$$

and similarly for $\widehat{K}_{ij}$. A product changes only if at least one of its two signs changes, and its change has magnitude at most 2. The low-margin implication above in Eq. (19) therefore gives

$$|\widehat{K}_{ij} - K_{ij}| \le \frac{2}{d}\sum_{q=1}^{d} [\mathbf{1}\{|(g_i)_q| \le \tau(\delta)\} + \mathbf{1}\{|(g_j)_q| \le \tau(\delta)\}] \le 4p_{\mathrm{aux}}\big(\tau(\delta)\big).$$

The same sign-stability argument holds for every target gradient. Applying the product bound to each auxiliary–target pair yields

$$|(\widehat{\beta}_r - \beta_r)_i| \le 2\left[p_{\mathrm{aux}}\big(\tau(\delta)\big) + p_{\mathrm{tar}}\big(\tau(\delta)\big)\right].$$

Since $\|\widehat{w}_\alpha\|_1 \le B$, the entrywise Gram bound implies

$$\left\|(\widehat{K} - K)\widehat{w}_\alpha\right\|_2 \le \sqrt{N}\max_{i,j\in[N]} |\widehat{K}_{ij} - K_{ij}|\,\|\widehat{w}_\alpha\|_1 \le 4B\sqrt{N}\,p_{\mathrm{aux}}\big(\tau(\delta)\big).$$

For every $\alpha \in \Delta_M$, convexity of the target combination gives

$$\left\|\widehat{\beta}_\alpha - \beta_\alpha\right\|_2 \le \sum_{r=1}^{M}\alpha_r\|\widehat{\beta}_r - \beta_r\|_2 \le 2\sqrt{N}\left[p_{\mathrm{aux}}\big(\tau(\delta)\big) + p_{\mathrm{tar}}\big(\tau(\delta)\big)\right].$$

Substituting these bounds into the residual decomposition Eq. (12) proves Eq. (16), simultaneously for all target preferences on the same event.

If $p(t) \le A_{\mathrm{mar}}t$ for $0 \le t \le t_0$, then, whenever m is large enough so that $\tau(\delta) \le t_0$,

$$\|\widehat{w}_\alpha - w^\star_\alpha\|_2 \le \frac{4(B+1)\sqrt{N}\,A_{\mathrm{mar}}C}{\mu}\sqrt{\frac{2}{m}\log\frac{4(N+M)d}{\delta}},$$

which gives the stated $\widetilde{O}(m^{-1/2})$ rate. If $p(0) > 0$, *i.e.,* at least one population gradient contains a nonzero fraction of exactly-zero coordinates, the bound in Eq. (16) need not vanish with m, and therefore does not by itself imply convergence of the SignGD valuations. □

We next compare signed and raw-gradient features with the population KMM solution induced by Adam's first update. We retain the same feasible set $\mathcal{W}_B$ and $\lambda = 1$.

Corollary B.7 (SignGD vs. first-step Adam MT-KMM)**.** *Consider Adam initialized with zero first and second moments. For a population gradient g, its normalized first-step feature in the zero-stabilizer limit is*

$$\phi^{\mathrm{Adam}}(g) := \lim_{\epsilon_{\mathrm{Adam}}\to 0} \frac{1}{\sqrt{d}} \frac{g}{|g| + \epsilon_{\mathrm{Adam}}} = \frac{1}{\sqrt{d}} \operatorname{sign}(g),$$

where division is coordinate-wise and $\operatorname{sign}(0) = 0$. Let $w^\star_{\alpha,\mathrm{Adam}}$ denote the population KMM minimizer using these features. Under the conditions of Thm. B.6, $w^\star_{\alpha,\mathrm{Adam}} = w^\star_\alpha$, and Eq. (16) bounds $\|\widehat{w}_\alpha - w^\star_{\alpha,\mathrm{Adam}}\|_2$ with probability at least $1-\delta$, simultaneously for every $\alpha \in \Delta_M$.

Proof. Bias correction gives $\widehat{m}_1 = g$ and $\widehat{v}_1 = g^2$, as derived in Sec. 4.2. Thus the limiting Adam and signed-gradient features coincide, so their population objectives and minimizers coincide. The bound follows directly from Thm. B.6. □

Corollary B.8 (Raw-gradient vs. first-step Adam MT-KMM)**.** *Use the same Adam reference as in Cor. B.7, and assume that its population auxiliary Gram matrix satisfies $\lambda_{\min}(K^{\mathrm{Adam}}) \geq \mu > 0$. Let $\widehat{w}_{\alpha,\mathrm{GD}}$ denote the empirical KMM minimizer using raw-gradient features $\phi^{\mathrm{GD}}(g) := g$. Suppose that the per-example raw gradients satisfy Assumption B.1, with $C \geq 1$. In the sample-size regime of Thm. B.5, with probability at least $1-\delta$, simultaneously for every $\alpha \in \Delta_M$,*

$$\|\widehat{w}_{\alpha,\mathrm{GD}} - w^\star_{\alpha,\mathrm{Adam}}\|_2 \leq \underbrace{\widetilde{O}\left(\frac{(B+1)\sqrt{N}}{\mu\sqrt{m}}\right)}_{\text{finite-preview estimation}} + \underbrace{O\left(\frac{(B+1)\sqrt{N}}{\mu}\right)}_{\text{population mismatch to Adam}}, \tag{20}$$

where $\widetilde{O}(\cdot)$ hides logarithmic factors in $N, M, d, 1/\delta$, and C is treated as a constant. The first term accounts for finite-preview estimation error, and the second bounds population mismatch to Adam.

Proof. Let $\varepsilon_{\mathrm{aux}}$ and $\varepsilon_{\mathrm{tar}}$ be the concentration bounds from Lems. B.3 and B.4, and define the population feature mismatch terms

$$b^{\mathrm{GD}}_{\mathrm{aux}} := \max_{i\in[N]} \|g_i - \phi^{\mathrm{Adam}}_i\|_2, \qquad b^{\mathrm{GD}}_{\mathrm{tar}} := \max_{r\in[M]} \|g_{\mathrm{tar},r} - \phi^{\mathrm{Adam}}_{\mathrm{tar},r}\|_2.$$

On the joint concentration event, which holds with probability at least $1-\delta$, the triangle inequality gives

$$\max_{i\in[N]} \|\widehat{g}_i - \phi^{\mathrm{Adam}}_i\|_2 \leq \varepsilon_{\mathrm{aux}} + b^{\mathrm{GD}}_{\mathrm{aux}}, \qquad \max_{r\in[M]} \|\widehat{g}_{\mathrm{tar},r} - \phi^{\mathrm{Adam}}_{\mathrm{tar},r}\|_2 \leq \varepsilon_{\mathrm{tar}} + b^{\mathrm{GD}}_{\mathrm{tar}}.$$

The target bound also holds for every convex combination $\alpha \in \Delta_M$. Raw empirical features have norm at most C, and Adam population features have norm at most $1 \leq C$. Applying the perturbation argument from the proof of Thm. B.5 to these total feature errors, with the Adam population objective, gives

$$\|\widehat{w}_{\alpha,\mathrm{GD}} - w^\star_{\alpha,\mathrm{Adam}}\|_2 \leq \frac{C\sqrt{N}}{\mu}\left[(2B+1)(\varepsilon_{\mathrm{aux}} + b^{\mathrm{GD}}_{\mathrm{aux}}) + \varepsilon_{\mathrm{tar}} + b^{\mathrm{GD}}_{\mathrm{tar}}\right].$$

The mismatch norms defining $b^{\mathrm{GD}}_{\mathrm{aux}}$ and $b^{\mathrm{GD}}_{\mathrm{tar}}$ can be evaluated exactly. Since $\phi^{\mathrm{Adam}}(g) = d^{-1/2}\operatorname{sign}(g)$, expanding the squared distance gives

$$b(g) = \|g - \phi^{\mathrm{Adam}}(g)\|_2^2 = \|g\|_2^2 + \frac{\|\operatorname{sign}(g)\|_0}{d} - \frac{2\|g\|_1}{\sqrt{d}}, \tag{21}$$

where $\|\operatorname{sign}(g)\|_0$ counts the nonzero coordinates of g. If all coordinates are nonzero, this reduces to $\|g\|_2^2 + 1 - 2\|g\|_1/\sqrt{d}$. Moreover, Assumption B.1 gives $\|g\|_2 \leq C$, so the triangle inequality yields

$$\|g - \phi^{\mathrm{Adam}}(g)\|_2 \leq \|g\|_2 + \|\phi^{\mathrm{Adam}}(g)\|_2 \leq C+1, \qquad b^{\mathrm{GD}}_{\mathrm{aux}},\, b^{\mathrm{GD}}_{\mathrm{tar}} \leq C+1.$$

Substituting the concentration bounds and these mismatch bounds into the perturbation bound above gives, on the same event and simultaneously for every $\alpha \in \Delta_M$,

$$\begin{aligned}\|\widehat{w}_{\alpha,\mathrm{GD}} - w^\star_{\alpha,\mathrm{Adam}}\|_2 \leq{}& \frac{\sqrt{8}\,C^2\sqrt{N}}{\mu\sqrt{m}}\left[(2B+1)\sqrt{L_{\mathrm{aux}}} + \sqrt{L_{\mathrm{tar}}}\right] \\ &+ \frac{4C^2\sqrt{N}}{3\mu m}\left[(2B+1)L_{\mathrm{aux}} + L_{\mathrm{tar}}\right] + \frac{2C(C+1)(B+1)\sqrt{N}}{\mu},\end{aligned} \tag{22}$$

where $L_{\mathrm{aux}} = \log(2N(d+1)/\delta)$ and $L_{\mathrm{tar}} = \log(2M(d+1)/\delta)$. In the regime $m \gtrsim \max\{L_{\mathrm{aux}}, L_{\mathrm{tar}}\}$, the m^{-1} terms are dominated by the $m^{-1/2}$ terms. Treating C as a constant and suppressing logarithmic factors yields Eq. (20). □

These corollaries separate finite-preview estimation error from population mismatch to the optimizer. For Adam's first step with zero moments and a vanishing stabilizer, SignGD has no mismatch term, whereas raw GD generally retains one even with exact population gradients. The sampling terms decay with m, while the worst-case mismatch contribution is independent of m.

B.3 Relation between the theoretical and implemented objectives

Retain $\lambda = 1$. Our numerical solver uses the penalized form

$$\widehat{w}_{\alpha,\gamma} \in \operatorname*{arg\,min}_{w\in\mathbb{R}^N} \left\{ \frac{1}{2} w^\top \widehat{K} w - \widehat{\beta}_\alpha^\top w + \gamma \|w\|_1 \right\}, \qquad \gamma = 0.05, \tag{23}$$

with the same γ throughout our experiments. Any solution also solves the constrained empirical problem over $\mathcal{W}_{B_{\alpha,\gamma}}$ at radius $B_{\alpha,\gamma} := \|\widehat{w}_{\alpha,\gamma}\|_1$. This radius is sample-dependent, not a predetermined score radius or the dataset-count limit k. We therefore compare directly with the population penalized solution at the same fixed γ:

$$w^\star_{\alpha,\gamma} \in \operatorname*{arg\,min}_{w\in\mathbb{R}^N} \left\{ \frac{1}{2} w^\top K w - \beta_\alpha^\top w + \gamma \|w\|_1 \right\}.$$

If $K \succeq \mu I$ with $\mu > 0$, the optimality conditions and monotonicity of the ℓ_1 subgradient imply a statement very similar to Eq. (12)

$$\|\widehat{w}_{\alpha,\gamma} - w^\star_{\alpha,\gamma}\|_2 \le \frac{1}{\mu} \left(\|(\widehat{K} - K)\widehat{w}_{\alpha,\gamma}\|_2 + \|\widehat{\beta}_\alpha - \beta_\alpha\|_2 \right).$$

Completing the square in the empirical objective and comparing with $w = 0$ also gives $\gamma\|\widehat{w}_{\alpha,\gamma}\|_1 \le \frac{1}{2}\|\widehat{\phi}_{\mathrm{tar},\alpha}\|_2^2$. Thus $\|\widehat{w}_{\alpha,\gamma}\|_1$ is uniformly bounded by $C^2/(2\gamma)$ for raw features and $1/(2\gamma)$ for SignGD. For fixed $\gamma > 0$, the entrywise concentration bounds used in Thms. B.5 and B.6 therefore yield, under their respective assumptions and with probability at least $1-\delta$, score error $\widetilde{O}(m^{-1/2})$ for raw gradients and $O(p(\tau(\delta)))$ for SignGD, simultaneously for every $\alpha \in \Delta_M$. The SignGD rate is $\widetilde{O}(m^{-1/2})$ under the same linear near-zero condition on p.

Note that all of these bounds concern scores *before mixture extraction*. After solving Eq. (23), we retain up to k largest positive scores and normalize the nonempty retained set as in Sec. 4.1, yielding the relative sampling weights used for training.

C Experimental setup

We provide the dataset inventory, baseline definitions, and training and evaluation settings for the experiments introduced in Sec. 5.1.

C.1 Targets and auxiliary datasets

Tab. 5 lists every dataset used in the paper: the eight targets summarized in Tab. 1a, CaseHOLD, which is not one of the eight targets and is used only in the three-target and RL panels (Secs. D.2, D.3 and E.1), and the N=96 auxiliary pool grouped by the task families previewed in Tab. 1b. All are publicly available and used under their respective licenses. The two- and three-target experiments (Secs. D.2 and E.1) use a 50-dataset subset of the full pool, the *small-target pool*, marked † in Tab. 5. Sec. D.9 uses the *preview-sweep pool*: the 50 datasets with more than 1024 examples, so that every preview size up to 1024 can be drawn in full.

Target data partitions. For each target we use a fixed split, unchanged throughout the paper. The 200 selection-validation examples used by DataModel, DataModel-CS and GradEx (Sec. C.2) are its last 200 rows, and each seed draws the t fine-tuning examples from the remaining rows, so the two never overlap. Evaluation uses a separate held-out split, capped at 800 examples per target. Two

target tasks also appear in the auxiliary pool, but their auxiliary counterparts come from *disjoint* splits; Sec. D.7 include the experiments removing these two from the auxiliary pool. Selection also uses labeled target examples beyond the t fine-tuning examples, but only to compute selection signals, never as training data. MT-KMM and One Step use 256 preview examples per target for gradient extraction, which overlap the t fine-tuning examples under 6%. One might argue that part of the gains comes from access to the additional labels on the 256 target preview examples, rather than from auxiliary selection. Sec. D.6 therefore trains target-only on its t examples together with the 256 preview examples to isolate the effect of the extra labels.

Table 4: Splits used for each target. Dashes mark targets absent from the auxiliary pool.

target	target split	auxiliary split	evaluation split	# eval
LEDGAR	`train`	`validation`	`test`	800
RACE (high)	`train`	—	`test`	800
QuAIL	`train`	—	`validation`	800
ReClor	`train`	—	`validation`	500
LogiQA2.0	`train`	—	`test`	800
CommonsenseQA	`train[:3000]`	`train[3000:]`	`validation`	800
ANLI	`train_r3`	—	`test_r3`	800
Banking77	`train`	—	`test`	800

Table 5: Complete dataset inventory. Auxiliary families are ordered target-related first; † marks the small-target pool and ‡ the GRPO pool. Datasets without an associated publication are identified by their Hugging Face repository.

dataset	HF repo (config)
Target tasks (8, plus CaseHOLD)	
LEDGAR (Tuggener et al., 2020)	`lex_glue (ledgar)`
RACE (high) (Lai et al., 2017)	`ehovy/race (high)`
QuAIL (Rogers et al., 2020)	`textmachinelab/quail`
ReClor (Yu et al., 2020b)	`metaeval/reclor`
LogiQA2.0 (Liu et al., 2023)	`datatune/LogiQA2.0`
CommonsenseQA (Talmor et al., 2019)	`tau/commonsense_qa`
ANLI (Nie et al., 2020)	`facebook/anli (r3)`
Banking77 (Casanueva et al., 2020)	`mteb/banking77`
CaseHOLD (Zheng et al., 2021)	`casehold/casehold`
Reading comprehension / QA (12)	
DROP† (Dua et al., 2019)	`ucinlp/drop`
RACE (middle)† (Lai et al., 2017)	`ehovy/race (middle)`
SQuAD† (Rajpurkar et al., 2016)	`rajpurkar/squad`
WebQuestions† (Berant et al., 2013)	`Stanford/web_questions`
NarrativeQA (Kočiský et al., 2018)	`deepmind/narrativeqa`
NewsQA (Trischler et al., 2017)	`lucadiliello/newsqa`
TriviaQA (Joshi et al., 2017)	`mandarjoshi/trivia_qa (rc)`
HotpotQA (Yang et al., 2018)	`hotpot_qa (distractor)`
CoQA (Reddy et al., 2019)	`stanfordnlp/coqa`
MultiRC (Khashabi et al., 2018)	`aps/super_glue (multirc)`
ReCoRD (Zhang et al., 2018)	`aps/super_glue (record)`
SQuAD 2.0 (Rajpurkar et al., 2018)	`rajpurkar/squad_v2`
Commonsense / science MC (13)	
ARC-Easy† (Clark et al., 2018)	`allenai/ai2_arc (ARC-Easy)`
BoolQ†‡ (Clark et al., 2019)	`google/boolq`
CommonsenseQA† (Talmor et al., 2019)	`tau/commonsense_qa`
HellaSwag† (Zellers et al., 2019)	`Rowan/hellaswag`
OpenBookQA† (Mihaylov et al., 2018)	`allenai/openbookqa`
PIQA† (Bisk et al., 2020)	`ybisk/piqa`
SciQ† (Welbl et al., 2017)	`allenai/sciq`
WinoGrande† (Sakaguchi et al., 2020)	`allenai/winogrande`
Social IQa‡ (Sap et al., 2019)	`allenai/social_i_qa`

Tab. 5 continued from the previous page.

dataset	HF repo (config)
SWAG (Zellers et al., 2018)	`allenai/swag (regular)`
COPA‡ (Roemmele et al., 2011)	`aps/super_glue (copa)`
QuaRTz (Tafjord et al., 2019)	`allenai/quartz`
WSC‡ (Levesque et al., 2011)	`aps/super_glue (wsc.fixed)`
Textual entailment / NLI (7)	
SNLI‡ (Bowman et al., 2015)	`stanfordnlp/snli`
MultiNLI‡ (Williams et al., 2018)	`nyu-mll/multi_nli`
WANLI‡ (Liu et al., 2022)	`alisawuffles/WANLI`
SciTail‡ (Khot et al., 2018)	`allenai/scitail (snli_format)`
QNLI‡ (Wang et al., 2018)	`nyu-mll/glue (qnli)`
RTE‡ (Dagan et al., 2006)	`nyu-mll/glue (rte)`
DocNLI‡ (Yin et al., 2021)	`saattrupdan/doc-nli`
Logical reasoning (4)	
ProofWriter‡ (Tafjord et al., 2021)	`renma/ProofWriter`
StrategyQA‡ (Geva et al., 2021)	`ChilleD/StrategyQA`
ARC-Challenge (Clark et al., 2018)	`allenai/ai2_arc (ARC-Challenge)`
ProntoQA‡ (Saparov & He, 2022)	`renma/prontoqa`
Intent / topic classification (6)	
CLINC-OOS (Larson et al., 2019)	`clinc_oos (plus)`
ATIS (Hemphill et al., 1990)	`tuetschek/atis`
SNIPS‡ (Coucke et al., 2018)	`benayas/snips`
AG News‡ (Zhang et al., 2015)	`ag_news`
DBpedia-14 (Zhang et al., 2015)	`dbpedia_14`
Yahoo Answers Topics (Zhang et al., 2015)	`yahoo_answers_topics`
Legal (8)	
BillSum† (Kornilova & Eidelman, 2019)	`billsum`
ECtHR (Task A)† (Chalkidis et al., 2022)	`lex_glue (ecthr_a)`
EUR-Lex† (Chalkidis et al., 2022)	`lex_glue (eurlex)`
LEDGAR† (Tuggener et al., 2020)	`lex_glue (ledgar)`
SCOTUS† (Chalkidis et al., 2022)	`lex_glue (scotus)`
UnfairToS† (Chalkidis et al., 2022)	`lex_glue (unfair_tos)`
MAUD (Wang et al., 2023)	`theatticusproject/maud`
PrivacyQA‡ (Ravichander et al., 2019)	`alzoubi36/privacy_qa`
Sentiment / text classification (5)	
SST-2‡ (Socher et al., 2013)	`nyu-mll/glue (sst2)`
IMDb‡ (Maas et al., 2011)	`stanfordnlp/imdb`
Amazon Polarity‡ (Zhang et al., 2015)	`fancyzhx/amazon_polarity`
Emotion‡ (Saravia et al., 2018)	`dair-ai/emotion`
TweetEval (sentiment)‡ (Barbieri et al., 2020)	`cardiffnlp/tweet_eval (sentiment)`
Mathematics (14)	
CAMEL-AI Math† (Li et al., 2023)	`camel-ai/math`
GSM8K (Socratic)† (Cobbe et al., 2021)	`openai/gsm8k (socratic)`
MATH-plus†‡ (Yue et al., 2024)	`TIGER-Lab/MATH-plus`
MathInstruct† (Yue et al., 2023)	`TIGER-Lab/MathInstruct`
MetaMathQA†‡ (Yu et al., 2023)	`meta-math/MetaMathQA`
OpenMathInstruct-2†‡ (Toshniwal et al., 2024)	`nvidia/OpenMathInstruct-2`
Orca-Math† (Mitra et al., 2024)	`microsoft/orca-math-word-problems-200k`
Tülu 3 Persona Math†‡ (Lambert et al., 2024)	`allenai/tulu-3-sft-personas-math-filtered`
Tülu 3 Persona Math (grade)† (Lambert et al., 2024)	`allenai/tulu-3-sft-personas-math-grade-filtered`
Tülu 3 Persona Algebra† (Lambert et al., 2024)	`allenai/tulu-3-sft-personas-algebra`
AQuA-RAT (Ling et al., 2017)	`deepmind/aqua_rat (raw)`
ASDiv‡ (Miao et al., 2020)	`EleutherAI/asdiv`
SVAMP‡ (Patel et al., 2021)	`ChilleD/SVAMP`

Tab. 5 continued from the previous page.

dataset	HF repo (config)
TheoremQA‡ (Chen et al., 2023)	`TIGER-Lab/TheoremQA`
Code (8)	
CodeAlpaca-20k† (Chaudhary, 2023)	`sahil2801/CodeAlpaca-20k`
Python Code Instructions†	`iamtarun/python_code_instructions_18k_alpaca`
Tülu 3 Persona Code† (Lambert et al., 2024)	`allenai/tulu-3-sft-personas-code`
MBPP (Austin et al., 2021)	`google-research-datasets/mbpp (full)`
CodeContests (Li et al., 2022)	`deepmind/code_contests`
LeetCode	`greengerong/leetcode`
SQL-Create-Context (b-mc2, 2023)	`b-mc2/sql-create-context`
Glaive Code Assistant	`glaiveai/glaive-code-assistant`
Medical (6)	
Medical Flashcards† (Han et al., 2023)	`medalpaca/medical_meadow_medical_flashcards`
MEDIQA† (Savery et al., 2020)	`medalpaca/medical_meadow_mediqa`
Medical MMLU†‡ (Han et al., 2023)	`medalpaca/medical_meadow_mmmlu`
PMC-LLaMA† (Wu et al., 2024)	`axiong/pmc_llama_instructions`
PubMed Causal†‡ (Yu et al., 2019)	`medalpaca/medical_meadow_pubmed_causal`
WikiDoc† (Han et al., 2023)	`medalpaca/medical_meadow_wikidoc`
Finance (3)	
FLARE-CFA†‡	`TheFinAI/flare-cfa`
Financial PhraseBank†‡ (Malo et al., 2014)	`AdaptLLM/finance-tasks (FPB)`
Financial Headlines†‡ (Sinha & Khandait, 2021)	`AdaptLLM/finance-tasks (Headline)`
General instruction following (10)	
Alpaca† (Taori et al., 2023)	`tatsu-lab/alpaca`
Dolly 15k† (Conover et al., 2023)	`databricks/databricks-dolly-15k`
OpenHermes-2.5† (Teknium, 2023)	`teknium/OpenHermes-2.5`
SlimOrca† (Lian et al., 2023)	`Open-Orca/SlimOrca`
UltraChat 200k† (Ding et al., 2023)	`HuggingFaceH4/ultrachat_200k`
Evol-Instruct† (Dissanayake et al., 2024)	`SurgeGlobal/evol-instruct`
WizardLM Evol-Instruct V2† (Xu et al., 2024)	`WizardLMTeam/WizardLM_evol_instruct_V2_196k`
Open-Platypus† (Lee et al., 2023)	`garage-bAInd/Open-Platypus`
Conifer† (Sun et al., 2024)	`ConiferLM/conifer`
smol-SmolTalk† (Allal et al., 2025)	`HuggingFaceTB/smol-smoltalk`

C.2 Baseline details

We extend the single-target baselines to select one auxiliary mixture for M targets by aggregating their target-specific signals. We write the aggregation using the preference weights $\alpha \in \Delta_M$ from Sec. 4.1; the main baseline comparisons use $\alpha_r = 1/M$. Other selection settings follow Sec. 5.1, and all methods share the final training and evaluation protocol of Sec. C.3. For GradEx and DataModel, scoring uses Llama-3.2-1B under SFT; the resulting auxiliary mixtures are reused at larger Llama scales.

Auxiliary allocation: uniform (u) and weighted (w). The labels (u) and (w) specify how the auxiliary training budget is allocated after selection. For a nonempty retained set S, uniform allocation assigns equal budget shares. Weighted allocation normalizes the retained positive dataset scores v_i:

$$\widetilde{w}_i^{(\mathrm{u})} = \begin{cases} 1/|S|, & i \in S, \\ 0, & i \notin S, \end{cases} \qquad \widetilde{w}_i^{(\mathrm{w})} = \begin{cases} \dfrac{v_i}{\sum_{j \in S} v_j}, & i \in S, \\ 0, & i \notin S. \end{cases}$$

Both allocations retain the k highest-scoring datasets; (w) additionally drops any with nonpositive scores. In all our selections every retained score is positive, so (u) and (w) keep the same datasets and differ only in allocation. We evaluate both allocations for DataModel, DataModel-CS, and

GradEx-RE; Tab. 2 reports their (w) variants. MT-KMM uses its postprocessed weighted mixture from Sec. 4.1.

One Step gradient alignment. We extend One Step by replacing the single-target representation with the composite $\widehat{\phi}_{\mathrm{tar},\alpha} = \sum_{r=1}^{M} \alpha_r \widehat{\phi}_{\mathrm{tar},r}$. Each auxiliary dataset is scored by cosine similarity to this composite, using the frozen-model gradient features of Sec. 5.1. One Step selects the top-k datasets and uses uniform allocation.

DataModel and DataModel-CS (Ilyas et al., 2022). Our multi-target extension replaces the single-target regression response with an aggregate of the responses on all M targets. Let $a_{j,r}$ denote target r's response for regression row j: validation accuracy for DataModel, or a paired accuracy difference for DataModel-CS. For each design, we scale each target's responses by their standard deviation across rows and combine them as

$$\widetilde{y}_j = \sum_{r=1}^{M} \alpha_r \frac{a_{j,r}}{\sigma_r + \varepsilon}, \qquad \sigma_r = \mathrm{std}_j\big(a_{j,r}\big),$$

where $\varepsilon > 0$ is a numerical stabilizer. This scaling reduces differences in response variation across targets before applying the target weights. We fit a LASSO regression of $\widetilde{y}_j$ on the subset-design vector x_j. Both fits use a LASSO coefficient of 10^{-3}. The fitted coefficients score auxiliary datasets for top-k selection.

Both designs train each scoring model on data from all targets together with an auxiliary subset. Write $\mathrm{acc}_r(S)$ for its validation accuracy on target r after training with auxiliary subset S. DataModel samples a random half of the pool, S_j, for each row. DataModel-CS, following the compressed-sensing variant of Zeng et al. (2026), draws $s_{j,i} \in \{+1, 0, -1\}$ independently with probabilities $\frac{1}{6}, \frac{2}{3}, \frac{1}{6}$ (Achlioptas, 2003), and uses two subsets, $S_j^+ = \{i : s_{j,i} \geq 0\}$ and $S_j^- = \{i : s_{j,i} \leq 0\}$, so zero-sign datasets appear in both. The design vectors and per-target responses are

$$\begin{aligned} \text{DataModel:} \quad & x_j = \mathbf{1}_{S_j}, && a_{j,r} = \mathrm{acc}_r(S_j), \\ \text{DataModel-CS:} \quad & x_j = \sqrt{3/N}\, s_j, && a_{j,r} = \mathrm{acc}_r(S_j^+) - \mathrm{acc}_r(S_j^-), \end{aligned}$$

where $\mathbf{1}_S$ is the subset indicator vector. Following Zeng et al. (2026), each design uses $10N = 960$ model trainings: 960 rows for DataModel and 480 paired rows for DataModel-CS. Scoring runs use the full auxiliary data in each subset rather than the final auxiliary training budget. Validation uses the last 200 examples of each target's training split, held out from training and separate from the final test set.

GradEx-FS and GradEx-RE (Li et al., 2024). We extend GradEx by aggregating its predicted improvements across targets to score each auxiliary subset. Let $\bar{g}_r$ be the mean projected gradient of the correct-answer log-odds on target r's validation examples, evaluated at a shared reference model. For a subset S, let $\Delta\theta_S$ denote the estimated parameter update in the projected space. The multi-target subset score is

$$\widehat{f}_\alpha(S) = \sum_{r=1}^{M} \alpha_r \langle \widehat{g}_r,\, \Delta\theta_S \rangle, \qquad \widehat{g}_r = \frac{\bar{g}_r}{\|\bar{g}_r\|_2 + \varepsilon}.$$

Normalizing each target gradient reduces the effect of differences in gradient magnitude on the aggregate score. Following Zeng et al. (2026, App. G.3), $\Delta\theta_S$ solves the logistic-regression surrogate on the subset's projected per-example gradients of the correct-answer log-odds, with every example weighted equally. We fit it with strong ℓ_2 regularization (inverse regularization strength 10^{-4}), implementing the single-class objective by flipping a random 30% of gradients and labelling them negative, which leaves the loss unchanged. We then rescale $\Delta\theta_S$ so that its norm in the full LoRA parameter space equals $0.1\|\theta^\star\|$, where $\theta^\star$ is the reference adapter. Validation uses the same last 200 rows of each target split as DataModel.

Both selectors use $\widehat{f}_\alpha$. GradEx-FS adds the dataset with the largest estimated marginal improvement at each step until k datasets are selected, then uses uniform allocation. GradEx-RE samples J subsets and scores dataset i by its mean subset score:

$$T_i = \frac{1}{n_i} \sum_{j:\, i \in S_j} \widehat{f}_\alpha(S_j), \qquad n_i = |\{j \in [J] : i \in S_j\}|.$$

It selects the top-k datasets. We use $J = 10N$ subsets following Li et al. (2024), with four datasets per subset.

The shared reference model is a rank-8 LoRA adapter on the q/k/v projections, trained on target and auxiliary data. We project the per-example log-odds gradients with a Rademacher matrix to $d = 100$, the dimension used by Zeng et al. (2026). Subset scores are computed on CPU, using auxiliary datasets as selection units.

Reference baselines and MT-KMM ablations. Target-only trains jointly on the target data without auxiliaries. Random selects k datasets uniformly at random and antiselect selects the k lowest-scored datasets under MT-KMM; both allocate the auxiliary budget equally. The no-K and selection-only ablations are reported in Tab. 3b. The no-K ablation uses the composite alignment scores directly:

$$\widehat{\beta}_\alpha = \widehat{\Phi}^\top \widehat{\phi}_{\mathrm{tar},\alpha}, \qquad S_\alpha = \mathrm{TopK}_+(\widehat{\beta}_\alpha).$$

It retains up to k datasets with the largest positive scores and applies (w) with $v_i = (\widehat{\beta}_\alpha)_i$, without using the auxiliary Gram matrix. Selection only retains $\mathrm{TopK}_+(\widehat{w}_\alpha)$ from MT-KMM and applies (u). Single-target KMM uses a one-hot preference vector to select for LEDGAR or Banking77 individually; its mixtures are evaluated on all eight targets in Tab. 7b.

C.3 Training and evaluation details

For both SFT and GRPO, the selected auxiliary examples are combined with the target training data. We convert allocation weights into numbers of examples: weighted allocation uses rounded counts, while uniform allocation divides the budget equally across datasets. Examples are sampled without replacement. If a dataset contains fewer examples than its assigned allocation, we use all available examples and do not redistribute the remaining budget to other datasets. We apply this procedure to all selection methods, so the realized auxiliary budget may be smaller than the specified budget.

Gradient extraction. For MT-KMM and One Step under SFT, each dataset is represented by 256 preview examples; Sec. D.9 varies this number. The preview draw and the LoRA initialization use a single fixed seed and are shared by every method, setting and seed. We average next-token cross-entropy gradients over these examples at the frozen reference model, with maximum sequence length 4096, taking gradients with respect to the LoRA parameters (rank 8, scaling parameter 16, q/k/v projections, dropout 0.05). For SignGD, we apply the sign transformation after averaging, with the normalization in Sec. 4.2. GradEx and GRPO use their own gradient features, described in Secs. C.2 and D.3.

Supervised fine-tuning. We train for one epoch using ms-swift 4.1.2 (Zhao et al., 2024) with five seeds. LoRA uses rank 8 and scaling parameter 32, and adapts all linear modules. We use fused AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$, and weight decay 0.1. The peak learning rate is 10^{-4}, with a cosine schedule and warmup ratio 0.05. Training uses a maximum sequence length of 4096 and bfloat16 precision, with TF32 disabled.

GRPO training. We train Llama-3.2-1B for one epoch with GRPO (Shao et al., 2024), using the `GRPOTrainer` of TRL 0.29.1 (von Werra et al., 2020) with the same LoRA configuration as SFT. Each prompt generates 8 completions at temperature 1.0, with at most 32 tokens per completion. Each optimizer step uses 8 prompts and their 64 completions on one GPU, with a per-device batch of 32 completions and two gradient accumulation steps. We use a peak learning rate of 10^{-5}, no KL penalty ($\beta = 0$), bfloat16 precision, and gradient checkpointing. The reward is 1.0 for an exact match to the gold answer plus 0.2 for an extractable answer. We extract answer letters for multiple-choice examples and numeric answers for mathematics examples. The target tasks and auxiliary pool are detailed in Sec. D.3.

Evaluation. All targets have finite answer sets. For each test example, we append each candidate answer to the prompt, sum its answer-token log-probabilities, and predict the candidate with the highest score. On Llama-3.1-70B, we reuse a prefix key-value cache to avoid recomputing the shared prompt for each candidate.

Batch sizes and hardware. All runs use H100 nodes with eight GPUs. For SFT, Llama-3.2-1B, Llama-3.1-8B, Qwen3-0.6B, and Qwen3-8B each use one GPU per run, with batch size 8 and two

gradient accumulation steps, giving an effective batch size of 16. Llama-3.1-70B and Qwen3-32B use eight GPUs under ZeRO-3 (Rajbhandari et al., 2020). Llama-3.1-70B uses per-device batch size 4 with four accumulation steps; Qwen3-32B uses per-device batch size 8 with two accumulation steps. Both have an effective batch size of 128.

D ADDITIONAL RESULTS

We provide supplementary results for Sec. 5.2, together with two controls and a pool-size sweep that the main text does not show. Unless stated otherwise, tables report mean held-out accuracy (%) over five seeds.

D.1 DETAILED BASELINE COMPARISONS

Tab. 6 reports per-target accuracy on Llama-3.2-1B. MT-KMM achieves the highest mean accuracy on five of the eight targets. All four DataModel variants improve over target-only training on LEDGAR and Banking77, but reduce accuracy on five targets. This pattern may reflect an auxiliary allocation that favors LEDGAR, where DataModel's gains are largest. ANLI is adversarially constructed, and every method stays near chance at 1B; with the same pipeline, MT-KMM reaches 46.7% at 70B.

Table 6: Per-target accuracy on Llama-3.2-1B, averaged over the eight target-size and auxiliary-budget settings in Sec. 5.1. The **mean** column is the unweighted average over targets, computed before rounding as in Tab. 2. **Bold** marks the best method in each column.

method	LEDGAR	RACE	QuAIL	Banking 77	Commonsense QA	ReClor	ANLI	LogiQA2.0	mean
target only	30.4	33.1	31.2	39.0	29.1	25.4	33.9	26.7	31.10
MT-KMM	50.2	**51.6**	**53.5**	43.2	**50.7**	**32.4**	33.5	**32.4**	**43.42**
One Step	51.2	41.7	47.2	43.8	42.7	27.7	33.9	28.0	39.51
GradEx-RE (u)	48.9	34.0	37.7	44.3	35.1	26.1	33.2	26.2	35.69
GradEx-RE (w)	48.6	34.9	39.3	44.8	37.5	26.1	33.3	26.6	36.39
GradEx-FS	31.4	38.5	41.5	45.6	38.0	27.0	**34.2**	28.4	35.55
DataModel (w)	52.7	28.5	30.5	45.4	27.2	25.4	32.8	25.2	33.47
DataModel (u)	51.9	28.6	30.0	**46.5**	26.4	25.5	32.9	25.1	33.37
DataModel-CS (w)	**54.1**	27.8	28.0	46.1	25.9	25.8	32.8	25.3	33.21
DataModel-CS (u)	53.7	28.2	28.0	45.1	24.3	25.6	32.9	25.3	32.86
random	35.9	32.1	31.9	43.9	31.7	25.9	32.9	26.3	32.57
antiselect	30.3	30.6	31.5	43.8	31.4	26.3	32.9	25.9	31.57

Tab. 7a reports the uniform counterparts to the weighted baselines in Tab. 2a. Weighting does not consistently improve these baselines: at Llama-3.1-70B, uniform allocation gives 0.48 and 0.50 points higher accuracy for GradEx-RE and DataModel-CS, respectively.

The single-target controls select auxiliaries for LEDGAR or Banking77 alone while retaining joint training and evaluation on all eight targets (Sec. C.2). On both Llama-3.2-1B and Qwen3-0.6B-Base, MT-KMM (Tab. 2a–b) has higher mean accuracy than either single-target control (Tab. 7b), supporting the use of all targets during selection.

Table 7: Additional baseline comparisons, averaged over each model's settings in Sec. 5.1. All methods are trained and evaluated jointly on the eight targets.

(a) Uniform allocation (u)

method	Llama-3.2-1B	Llama-3.1-8B	Llama-3.1-70B
GradEx-RE (u)	35.69	58.90	65.32
DataModel (u)	33.37	56.02	64.46
DataModel-CS (u)	32.86	55.62	64.85

(b) Single-target KMM selection

model	LEDGAR	Banking77
Llama-3.2-1B	36.79	32.64
Qwen3-0.6B-Base	43.81	43.66

D.2 Additional Qwen results

Model-scale comparisons are reported in Tab. 2b. Tab. 8 compares three target configurations on Qwen3-0.6B-Base. The two-target set contains LEDGAR and RACE; the three-target set adds CaseHOLD, which is absent from the eight-target benchmark. The smaller target sets use the small-target pool (Sec. C.1). The pool size, k, and auxiliary budgets also differ between the smaller target sets and the eight-target setting, so these results do not isolate the effect of target count. MT-KMM has the highest mean accuracy in each configuration.

Table 8: Accuracy across target configurations on Qwen3-0.6B-Base.

targets	pool / k	aux budget	target only	random	antiselect	One Step	MT-KMM
2 (LEDGAR, RACE)	50 / 5	{800, 1600}	36.70	39.31	39.47	45.72	**48.87**
3 (+CaseHOLD)	50 / 5	{800, 1600}	38.60	41.18	38.28	45.35	**49.23**
8 (full set)	96 / 10	{3200, 6400}	40.40	43.55	43.70	45.81	**46.60**

Tab. 9 reports the Qwen component ablations, complementing the Llama results in Tab. 3b. Removing K lowers mean accuracy by 0.57–0.76 points across Qwen models, a smaller decrease than in the Llama ablations. The Qwen3-0.6B comparison has $p = 0.0026$ (Tab. 24).

Table 9: Qwen component ablations under the settings of Tab. 2b. n/a indicates an unreported configuration.

model	selection only	no K	MT-KMM
Qwen3-0.6B-Base	45.57	45.84	**46.60**
Qwen3-8B-Base	n/a	68.64	**69.38**
Qwen3-32B	n/a	68.82	**69.39**

D.3 Reinforcement learning with GRPO

We use Llama-3.2-1B with QuAIL, RACE, CommonsenseQA and CaseHOLD as targets. GRPO needs a reward that the base model sometimes earns. LEDGAR and Banking77 fail this: their 100 and 77 labels are not listed in the prompt, so every completion scores zero. Of the remaining targets, we use the four with the most headroom in preliminary runs. The auxiliary pool contains the 34 datasets of the full pool whose answers can be scored automatically, excluding copies of the targets, rewritten as lettered multiple choice or as mathematics with a numeric answer, and marked ‡ in Tab. 5. Because GRPO learns only from reward differences among the completions for one prompt, we also drop datasets on which the base model gives all 8 completions the same reward for more than 85% of 32 prompts. Selection uses signed, group-normalized GRPO policy gradients computed at the frozen initialization. Training and reward details are in Sec. C.3.

Tab. 10 reports the no-K and selection-only ablations defined in Sec. C.2. Under GRPO, equal allocation reduces mean accuracy by 1.68 points, while the no-K ablation reduces it by 0.83 points.

Both components therefore contribute in this regime, with a larger observed decrease from removing the learned allocation weights.

Table 10: GRPO component ablations on the four targets listed above, averaged over all combinations of target sizes and auxiliary budgets.

target examples per task	k	aux budget	selection only	no K	MT-KMM
$\{20, 40, 100, 200\}$	5	$\{800, 1600\}$	40.65	41.50	**42.33**

D.4 Number of targets

Tab. 11 gives the absolute accuracies for Fig. 2a. The nested target sets start with LEDGAR and RACE, add QuAIL at $M = 3$, Banking77 and CommonsenseQA at $M = 5$, and ReClor, ANLI, and LogiQA2.0 at $M = 8$.

Table 11: Accuracy for the target-count sweep of Fig. 2a, using the two auxiliary-budget schedules defined there.

		fixed total				fixed per-target			
targets	target only	MT-KMM	One Step	random	antiselect	MT-KMM	One Step	random	antiselect
2	25.74	**54.44**	43.14	31.74	28.54	**47.54**	37.84	30.84	26.74
3	27.32	**53.62**	44.22	31.22	28.02	**49.32**	41.32	30.62	27.32
5	31.72	**49.82**	46.32	34.42	35.32	**47.92**	43.92	34.12	35.52
8	31.10	**43.42**	39.51	32.57	31.57	**43.42**	39.51	32.57	31.57

D.5 Target size and auxiliary budget

Tabs. 12 and 13 provide the numerical results for Fig. 2b-c, respectively. The target-size table also includes random selection with 6400 auxiliary examples.

Table 12: Accuracy by target size at two auxiliary budgets.

		aux 3200		aux 6400	
target examples per task	target only	random	MT-KMM	random	MT-KMM
20	26.3	29.3	42.0	29.7	**43.3**
40	31.7	30.3	42.5	30.8	**42.6**
100	31.9	33.2	43.1	33.8	**43.8**
200	34.6	37.3	**45.5**	36.2	44.8
400	38.3	39.4	46.2	39.8	**47.3**
1000	47.6	45.2	49.6	45.1	**50.5**
2000	51.6	49.1	51.3	48.8	**53.4**

Table 13: Accuracy with 200 target examples per task. Column headings are nominal auxiliary budgets. MT-KMM uses 3105, 5876, and 10184 auxiliary examples at budgets 3200, 6400, and 12800, respectively, due to limited data in the selected datasets.

method	200	400	800	1600	3200	6400	12800
target only	34.6	34.6	34.6	34.6	34.6	34.6	34.6
random	35.1	33.6	38.2	36.9	37.3	36.2	36.1
MT-KMM	**36.0**	**38.8**	**39.8**	**42.0**	**45.5**	**44.8**	**47.0**

D.6 Controlling for preview labels

MT-KMM and One Step use 256 labeled preview examples per target to compute selection gradients, but never train on them (Sec. C.1). To test whether the gains come from these extra labels rather

than from auxiliary selection, we train target-only on its t examples together with the 256 preview examples, with no auxiliary data. MT-KMM still exceeds this baseline at every target size, by 6.1 and 6.4 points on average at the two auxiliary budgets.

Table 14: Target-only trained on t or $t + 256$ target examples, and MT-KMM, which trains on the t examples plus auxiliary data.

	target only		MT-KMM	
	t	$t + 256$	aux 3200	aux 6400
$t = 20$	26.3	34.9	42.0	**43.3**
$t = 40$	31.7	33.5	42.5	**42.6**
$t = 100$	31.9	38.5	43.1	**43.8**
$t = 200$	34.6	41.9	**45.5**	44.8
mean	31.1	37.2	43.3	**43.6**

D.7 Controlling for same-target auxiliaries

LEDGAR and CommonsenseQA also appear in the auxiliary pool, drawn from splits disjoint from their target data (Sec. C.1). Part of the gain could therefore come from more data from the target's own distribution rather than from transfer across datasets. In the reference selection, these two datasets receive 17.1% of the auxiliary budget on Llama and 11.8% on Qwen3. To test this, we remove both from the pool and repeat the reference panel. MT-KMM remains the best method on both models.

Table 15: Accuracy with the full pool (96) and with LEDGAR and CommonsenseQA removed (94).

model	pool	target only	random	One Step	MT-KMM
Llama-3.2-1B	96	31.10	32.57	39.51	**43.42**
	94	31.10	32.32	37.98	**41.32**
Qwen3-0.6B-Base	96	40.40	43.55	45.81	**46.60**
	94	40.40	43.24	44.07	**44.53**

D.8 Auxiliary pool size

Tab. 16 varies the auxiliary pool size under the reference settings in Sec. 5.1. We use nested subsets of the 96-dataset pool sampled with a fixed seed, with one pool draw per size. MT-KMM achieves the highest mean accuracy at every tested pool size.

Table 16: Accuracy by auxiliary pool size at $k = 10$.

pool	target only	random	antiselect	One Step	MT-KMM
15	31.10	32.08	30.49	33.12	**34.58**
30	31.10	31.71	30.46	33.99	**34.82**
60	31.10	31.96	31.57	37.38	**41.44**
96	31.10	32.57	31.57	39.51	**43.42**

MT-KMM's largest increase is 6.62 points between pools of 30 and 60 datasets, while random selection changes by 0.25 points. With one pool draw per size, this sweep does not separate the effect of a larger candidate set from that of particular added datasets.

D.9 Gradient sample size

Tab. 17 supplements Fig. 3a with top-4 overlap and allocation changes. We use Llama-3.2-1B with the eight targets and the preview-sweep pool (Sec. C.1), so the pool differs from the reference setup in Sec. 5.1. Each selection is compared with the $m = 256$ selection from the same draw. Top-10 and

top-4 overlap count shared datasets; budget moved is half the ℓ_1 distance between the auxiliary allocation vectors, expressed as a percentage. Accuracy is reported as the mean and standard deviation over ten selection draws with the training seed held fixed.

Table 17: Selection stability and accuracy as the number of gradient samples m varies, over ten selection draws per m.

m	top-10 overlap	top-4 overlap	budget moved (%)	accuracy (%) $\pm$ std
4	5.1	2.8	28.6	43.97 ± 0.90
8	6.3	3.0	21.8	43.95 ± 1.00
16	7.5	2.9	16.9	44.01 ± 0.62
32	8.7	3.5	10.5	43.50 ± 0.41
64	9.2	3.8	6.6	43.75 ± 0.50
128	9.7	3.8	3.8	43.74 ± 0.44
256	10.0	4.0	0.0	43.38 ± 0.36
512	9.8	3.9	2.9	43.31 ± 0.22
1024	9.8	3.8	3.5	43.23 ± 0.29

D.10 Selection cost and scalability

Tab. 18 reports selection costs for Llama-3.2-1B on H100 hardware. GPU-hours cover model computation required by the selector, including validation, and exclude idle time and downstream training and evaluation. CPU work is reported separately below.

Measured costs. MT-KMM's GPU work consists of gradient extraction under the reference settings in Sec. 5.1. GradEx trains one shared reference model and extracts per-example gradients; its FS and RE variants reuse these features. DataModel and DataModel-CS each require $10N = 960$ model trainings, corresponding to 960 unpaired rows and 480 paired rows, respectively (Sec. C.2). Each training takes about 85 minutes for DataModel or 141 for DataModel-CS on a single H100, followed by 0.25 GPU-hours for validation across the eight targets. Tab. 18 reports measured totals; multiplying the average time per training by the number of trainings gives slightly smaller numbers.

CPU work. Gram construction, the quadratic program, and allocation postprocessing run on CPU and finish within seconds at the reference setup. Gram construction costs $O(N^2 d)$ for gradient dimension d; the reported time is 0.13 seconds at $N = 96$, measured with 8 threads on a 48-vCPU Intel Xeon Platinum instance, and about 14 seconds at $N = 1000$, estimated from the $O(N^2 d)$ scaling.

Extrapolation. With a fixed preview size, MT-KMM extracts gradients once for each auxiliary and target dataset. GradEx trains one shared reference model and extracts reusable per-example gradients; with fixed training epochs and data amounts per dataset, both operations grow linearly with the auxiliary and target dataset counts. Holding model size and sequence lengths fixed, both methods therefore require $O(N + M)$ GPU work, excluding CPU selection costs. For MT-KMM and GradEx, the projections in Tab. 19 scale the measured GPU cost with $N + M$, assuming a common average cost per dataset for targets and auxiliaries. For both DataModel designs, we scale per-model training time with auxiliary subset size, add 0.031 GPU-hours per target for validation, and multiply by $10N$ trainings. New auxiliary datasets are assumed to contain about 3500 examples.

Table 18: Measured selection cost at $N = 96$ and $M = 8$ using Llama-3.2-1B on an 8×H100 node. Downstream training and evaluation are excluded.

method	model trainings	GPU-hours	vs MT-KMM
MT-KMM	0	≈ 0.7	$1\times$
GradEx (all 3 variants)	1	≈ 9.7	$\sim 14\times$
DataModel	$10N$=960	≈ 1600	$\sim 2300\times$
DataModel-CS	$10N$=960	≈ 2530	$\sim 3600\times$

Table 19: Measured and extrapolated selection cost in GPU-hours as the number of targets or auxiliary datasets varies.

(a) vs number of targets M (pool N=96)

		measured	estimated		
method	trainings	M=8	M=100	M=1000	vs MT-KMM at M=1000
MT-KMM	**0**	0.7	∼1.3	∼7	1×
GradEx	**1**	9.7	∼18	∼100	∼14×
DataModel	$10N$	∼1600	∼4400	∼31000	∼4400×
DataModel-CS	$10N$	∼2530	∼5300	∼32000	∼4600×

(b) vs pool size N (targets M=8)

		measured	estimated		
method	trainings	N=96	N=200	N=1000	vs MT-KMM at N=1000
MT-KMM	**0**	0.7	∼1.4	∼7	1×
GradEx	**1**	9.7	∼19	∼95	∼14×
DataModel	$10N$	∼1600	∼6400	∼150000	∼21000×
DataModel-CS	$10N$	∼2530	∼10000	∼250000	∼36000×

E Additional analysis

We provide supplementary details for Sec. 5.3, followed by gradient-source comparisons and statistical tests. Unless stated otherwise, tables report mean held-out accuracy (%) over five seeds.

E.1 Target preference and auxiliary allocation

Tabs. 20 and 21 give accuracies and dataset-level allocations for the preference experiments in Sec. 5.3. We use Llama-3.2-1B with SignGD, the small-target pool (Sec. C.1), and $k = 5$, with $t \in \{20, 40, 100, 200\}$ and auxiliary budgets $\{800, 1600\}$. The three-target setting adds CaseHOLD to LEDGAR and RACE. Uniform preferences weight all targets equally; each three-target *target-heavy* setting assigns 0.6 to the named target and 0.2 to each other. The two allocation rules are defined in Sec. C.2.

Table 20: Accuracy under different target preferences, averaged over eight target-size and auxiliary-budget settings. Overall is the unweighted mean over targets. **Bold** marks the best value in each column.

(a) 2 targets

	selection and allocation			selection only		
composite	LEDGAR	RACE	overall	LEDGAR	RACE	overall
RACE-heavy (α=0.3)	48.91	**43.06**	45.98	41.38	37.38	39.38
uniform (α=0.5)	52.72	42.27	47.49	41.79	**37.46**	39.62
LEDGAR-heavy (α=0.7)	**53.94**	41.85	**47.89**	**42.37**	37.17	**39.77**

(b) 3 targets

	selection and allocation				selection only			
composite	LEDGAR	RACE	CaseHOLD	overall	LEDGAR	RACE	CaseHOLD	overall
uniform	47.94	42.97	**40.55**	**43.82**	41.69	38.21	27.00	35.64
LEDGAR-heavy	**51.68**	41.53	35.71	42.97	**43.14**	39.01	**29.97**	**37.37**
RACE-heavy	45.81	**44.45**	39.67	43.31	42.04	**39.03**	27.95	36.34
CaseHOLD-heavy	46.42	43.91	40.11	43.48	41.99	38.94	27.92	36.28

In the three-target experiment, LEDGAR and RACE reach their highest mean accuracy under their own preferences for both allocation rules, whereas CaseHOLD peaks under uniform preferences

with weighted allocation and under LEDGAR emphasis with equal allocation. The preference weights therefore steer which auxiliary data is selected (Tab. 21), but do not guarantee downstream performance on the emphasized target (Tab. 20). Fig. 5 aggregates the three-target allocations in Tab. 21b by dataset type.

Table 21: Auxiliary training budget shares (%) among the top-5 selected datasets. Shares are allocation weights; realized fractions of auxiliary rows differ by at most 0.2 points. Dashes indicate unselected datasets; **bold** marks the largest share in each column. Shares may not sum to 100% due to rounding.

(a) 2 targets

domain	dataset	RACE-heavy	uniform	LEDGAR-heavy
reading comp.	RACE (middle)	**70.5**	**49.6**	28.4
legal	LEDGAR	27.3	47.9	**67.3**
reading comp.	SQuAD	0.6	1.4	1.8
commonsense	WinoGrande	–	–	1.6
general	SlimOrca	0.5	1.0	0.9
medical	MEDIQA	1.1	0.1	–

(b) 3 targets

domain	dataset	uniform	LEDGAR-heavy	RACE-heavy	CaseHOLD-heavy
reading comp.	RACE (middle)	**55.8**	30.4	**75.9**	**63.1**
legal	LEDGAR	37.0	**64.0**	19.3	25.4
general	SlimOrca	2.6	2.2	1.4	3.9
reading comp.	SQuAD	2.5	2.8	1.1	3.5
medical	MEDIQA	2.2	–	2.4	4.1
commonsense	BoolQ	–	0.6	–	–

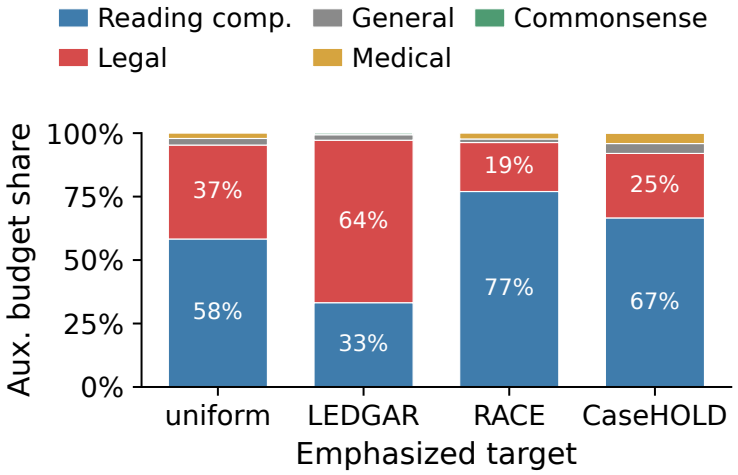

Figure 5: Auxiliary budget shares for the three-target preferences in Tab. 21b, aggregated by dataset type.

E.2 Gradient source and model transfer

Tab. 22 compares the reference gradient source in Sec. 5.1 with an 8B model from the same family. For each source, we compute all 96 auxiliary and 8 target gradients, solve MT-KMM, and train on the resulting mixture. The rank-32 row changes the LoRA rank for gradient extraction only; downstream training keeps rank 8. Random selection uses no gradients, so it is shown once.

The effect of the gradient source depends on the model family: the smaller source gives higher mean accuracy in the tested Llama configurations, whereas the 8B source improves Qwen3-8B by 0.54 points; all source differences are below 1.5 points.

Table 22: Gradient-source comparison. The default configuration uses Llama-3.1-8B for training, SignGD features, and rank-8 LoRA. **Bold** marks the higher mean for each source pair.

configuration	target only	small source	8B source	Δ	p
default	54.40	**61.63**	60.41	-1.22	0.0001
LoRA rank 32	54.40	**61.78**	60.44	-1.34	0.0001
Train Llama-3.2-1B	31.10	**43.42**	42.49	-0.93	<0.0001
Qwen3-8B	64.40	69.38	**69.92**	$+0.54$	0.0022
random	54.40	54.77		–	–

Small source: Llama-3.2-1B, or Qwen3-0.6B for the Qwen row.
Δ is 8B minus small source, in percentage points; p compares the two sources (Sec. E.3).

E.3 Statistical significance

Sources of randomness. The preview examples and the LoRA initialization used for gradient extraction are drawn once at a fixed seed, so each method has a single selection; Sec. D.9 measures the sensitivity to the preview draw. Each of the five seeds redraws the target training examples, the auxiliary examples sampled into the blend, and the training initialization, in the same way for every method, so runs with the same seed are paired across methods.

Test. Our conclusions concern the target-size and auxiliary-budget settings we evaluate (eight per model, five for Llama-3.1-70B); we make no claim about untested settings. For each seed, we average accuracy over targets and settings, and compare MT-KMM with each baseline by a two-sided paired t-test on the five seed means, with 4 degrees of freedom and 95% t-intervals. Because each seed's settings are averaged before testing, the test remains valid under any dependence among settings within a seed; it assumes that the seed-level differences are independent and approximately normal. With five seeds, a permutation test cannot reach $p < 0.05$, so we use the t-test. Because our claim is that MT-KMM exceeds every baseline, each comparison must hold on its own, so Tabs. 24 to 26 report unadjusted p-values; as a robustness check, we also apply Holm's step-down correction across their 58 comparisons.

Results. MT-KMM has higher accuracy than the baseline in all 58 comparisons, with $p < 0.05$ in each (largest $p = 0.0165$). The conclusion is unchanged after Holm correction (largest adjusted $p = 0.0173$). To check consistency across settings, we also apply the paired t-test to the five seeds within each setting. MT-KMM has the higher mean in 420 of 428 setting-level comparisons, and $p < 0.05$ without correction in 385. The eight exceptions are against One Step (5 of 8 settings), no K (6 of 8) and selection only (7 of 8) on Qwen3-0.6B-Base, against One Step on Qwen3-8B-Base (7 of 8), and against no K under GRPO (7 of 8). Tab. 22 compares gradient sources with the same seed-level test, as a separate family.

Table 23: MT-KMM mean accuracy and the half-width of a 95% t-interval over its five seed means, each averaged over targets and settings; the interval reflects variation across seeds.

model	MT-KMM
Llama-3.2-1B	43.42 ± 0.33
Llama-3.1-8B	61.63 ± 0.14
Llama-3.1-70B	69.38 ± 0.20
Qwen3-0.6B-Base	46.60 ± 0.26
Qwen3-8B-Base	69.38 ± 0.12
Qwen3-32B	69.39 ± 0.18
Llama-3.2-1B (GRPO)	42.33 ± 0.47

Table 24: Two-sided paired p-values of MT-KMM against common baselines and component ablations. Accuracies are in Tabs. 2, 3, 9 and 10.

model	random	antiselect	One Step	selection only	no K
Llama-3.2-1B	0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Llama-3.1-8B	0.0005	<0.0001	<0.0001	<0.0001	<0.0001
Llama-3.1-70B	0.0032	<0.0001	<0.0001	<0.0001	0.0002
Qwen3-0.6B-Base	0.0043	0.0004	0.0049	0.0003	0.0026
Qwen3-8B-Base	0.0165	<0.0001	0.0003	n/a	<0.0001
Qwen3-32B	0.0001	0.0001	0.0007	n/a	0.0001
Llama-3.2-1B (GRPO)	0.0006	<0.0001	0.0009	0.0001	0.0069

Table 25: Two-sided paired p-values of MT-KMM against GradEx and DataModel. Accuracies are in Tab. 2a and Tab. 7a; allocation labels are defined in Sec. C.2.

method	Llama-3.2-1B	Llama-3.1-8B	Llama-3.1-70B
GradEx-RE (u)	<0.0001	<0.0001	<0.0001
GradEx-RE (w)	<0.0001	<0.0001	<0.0001
GradEx-FS	<0.0001	<0.0001	<0.0001
DataModel (w)	<0.0001	<0.0001	<0.0001
DataModel (u)	<0.0001	<0.0001	<0.0001
DataModel-CS (w)	<0.0001	<0.0001	<0.0001
DataModel-CS (u)	<0.0001	<0.0001	<0.0001

Table 26: Two-sided paired p-values of MT-KMM against single-target KMM selection on the eight-target benchmark. Accuracies are in Tab. 7b.

model	Single-target KMM (LEDGAR)	Single-target KMM (Banking77)
Llama-3.2-1B	<0.0001	<0.0001
Qwen3-0.6B-Base	<0.0001	<0.0001