

CLIVO: DIFFERENTIABLE CLUSTER-BASED INSTRUCTION MIXING WITH LEARNED VALIDATION OBJECTIVES

Yuhan Wang¹ Rui Hu¹ Batian Di² Tan Jiang³
 Dongsheng Ma¹ Leichao Dong⁴ Yuanzi Li² Yalin Feng⁵

¹Peking University ²Renmin University of China ³Zhejiang University
⁴Xi'an Jiaotong University ⁵Nanyang Technological University

ABSTRACT

Target-aware multimodal instruction selection requires useful examples and an appropriate balance of supervision. Sample-level rankings can overrepresent similar content without optimizing this balance; group-level methods face coarse units, predefined set-level validation signals, and costly or inaccurate mixture search. We propose CLIVO, a target-aware framework connecting fine-grained semantic allocation, automatically learned validation objectives, and differentiable mixture search. Specifically, semantic clusters provide independently adjustable shares at the hundred-cluster scale. Correlations between per-example merged-proxy losses and real probe performance identify target-relevant validation examples; learned nonnegative weights combine their losses into an objective. A single backward pass through the merged proxy supplies gradients for all shares, avoiding an additional mixture-response regression model. Selecting 100K instructions from an 8M pool, CLIVO attains 97.2% of the full-data four-target average score with only 1.25% of the data. It improves on the strongest evaluated selection baseline by 1.16 points using random within-cluster sampling; combining learned quotas with sample ranking increases this advantage to 1.55 points. Analyses show that finer units benefit most targets, short-trained experts preserve mixture responses, sample weighting retains useful validation information, and direct search consistently reduces the learned objective. CLIVO also supports interpretable allocation, modular selection, and reuse across targets and granularities.

1 INTRODUCTION

Vision-language models (VLMs) have demonstrated strong capabilities in visual understanding, document parsing, multimodal reasoning, and interactive tasks (Comanici et al., 2025; Wang et al., 2025; Bai et al., 2025a; Hong et al., 2025; Huang et al., 2026; Team et al., 2026). Their development combines multimodal representation pretraining with instruction-following post-training (Yin et al., 2025; Team et al., 2025b;a; Radhakrishna et al., 2025; Lin et al., 2024). Multimodal instruction tuning develops instruction following through instruction-response examples about visual inputs (Liu et al., 2023a; Dai et al., 2023; Du et al., 2025). The training data therefore shape both the supervision a model receives and the tasks it practices. Effective tuning depends on supervision quality (Luo et al., 2025) as well as the composition of the training set (Zhang et al., 2025). Large instruction pools offer broad coverage of tasks and response formats (Xu et al., 2024), yet this breadth does not ensure that every example is useful: incorrect responses and hallucinated content can degrade model performance (Gou et al., 2025). This motivates **data selection**, which concentrates training on useful supervision. Prior work shows that carefully selected subsets can match full-data performance and even surpass it on individual benchmarks (Liu et al., 2025). Compact subsets also support repeated adaptation to different applications. A document assistant, for example, needs a different balance of text reading, visual recognition, and reasoning from a general visual assistant. Among useful examples, relative abundance determines how much training each type of supervision receives. Selection must therefore address both which examples to retain and how to balance their contribution to the intended capability. At a fixed training-data budget, we ask:

How should instruction data be mixed to best support a target capability?

Existing data selection methods operate at the sample or group level. *Sample-level methods* rank examples using signals such as gradient influence in ICONS (Wu et al., 2024), necessity in MLLM-Selector (Ma et al., 2025), and importance and diversity in CoIDO (Yan et al., 2026). However, selecting the highest-scoring examples can overrepresent similar content, reducing diversity and degrading downstream performance (He et al., 2025). Diversity-aware scores help, but often leave content shares implicit. *Group-level methods* explicitly allocate shares across sources, domains, or semantic groups, requiring choices of allocation **unit**, evaluation **signal**, and mixture **learning** rule.

Unit: fine-grained allocation remains difficult to search. Predefined source shares tie heterogeneous instructions together. Semantic clusters offer finer control, but COINCIDE’s density and transferability heuristics (Lee et al., 2024) do not optimize measured target performance. CLIMB (Diao et al., 2025) uses downstream feedback to search 21 groups in its main setting, with sampling costs limiting substantially finer search.

Signal: validation relies on predefined sets and coarse loss averages. Costly candidate training motivates proxy validation losses. CLIMB uses predefined task sets, while CAMEL (Li et al., 2026a) maps validation losses to benchmark scores. Neither automatically selects target-relevant examples, and their set-level averages can obscure informative differences between individual responses.

Learning: search is costly and susceptible to response-prediction error. DoReMi (Xie et al., 2023a) couples reweighting to proxy training, limiting reuse across targets and groupings. RegMix (Liu et al., 2024b), DeMix (Li et al., 2026c), and MergeMix (Wang et al., 2026) instead fit response predictors from evaluated mixtures. High-dimensional candidate labeling is costly, and selecting predicted optima can amplify errors on unseen mixtures (Smith & Winkler, 2006).

We propose **CLIVO** (CLuster-based Instruction Mixing with Learned Validation Objectives) to jointly address these challenges. Specifically, **fine-grained semantic units** provide adjustable shares at the hundred-cluster scale. A **learned validation signal** selects examples using proxy-loss correlations with real probe scores and fits nonnegative weights to aggregate their losses. Expert merging supplies proxies without retraining for each mixture (Berasi et al., 2026). Finally, **gradient-based mixture learning** differentiates this objective through fixed experts to update all shares jointly, without an additional mixture-response predictor.

Using 100K instructions (1.25% of the pool) and random within-cluster sampling, CLIVO attains 97.2% of the full-data four-target average score. It improves on the strongest evaluated data selection baseline at this budget by 1.16 points; within-quota ranking increases the advantage to 1.55 points. Our analyses show that finer units benefit most targets, moderate-budget experts and sample weighting provide reliable feedback while preserving individual responses, and direct gradient search lowers every target’s learned objective.

Importantly, CLIVO also supports **interpretable, flexible, and reusable data curation**. Semantic visualizations expose content choices; modular encoders, objectives, and selectors accommodate different applications. With the base and partition fixed, experts, probes, and cached losses support new targets. A single backward pass supplies gradients for all shares even at finer granularities, avoiding an additional higher-dimensional regression fit.

Our contributions are:

- **A unified target-aware framework.** We connect fine-grained semantic allocation, automatic validation-example selection and weighting, and differentiable mixture search through reusable expert proxies.
- **Evidence for fine-grained allocation, reliable signals, and direct search.** We demonstrate benefits from finer units on most targets, reliable feedback from moderate-budget experts and individual validation responses, and objective improvement through direct search for every target.
- **Interpretable, flexible, and reusable curation.** We make allocations interpretable through visualization, demonstrate flexible combinations with sample selectors, and reuse experts and feedback across targets and granularities.

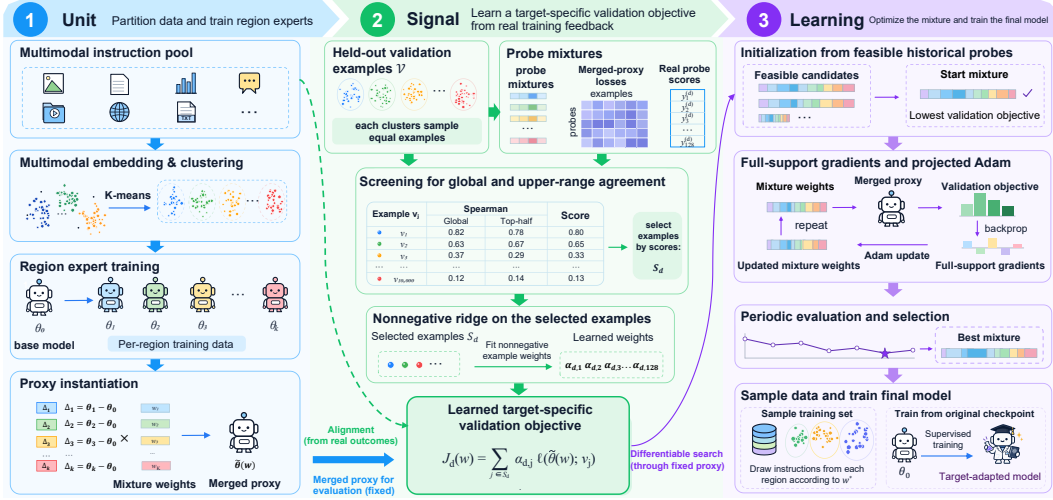


Figure 1: **CLIVO overview: learning what to train on and how to evaluate the mixture.** (1) **Unit:** a fixed multimodal encoder and K -means organize the instruction pool into semantic clusters with adjustable training shares. (2) **Signal:** probe models are trained on sampled mixtures and evaluated on downstream benchmarks. Cluster experts are merged using the same shares to construct proxies. Correlations between per-example proxy losses and real scores identify target-relevant validation examples; learned nonnegative weights combine their losses into an objective. (3) **Learning:** with experts and validation weights frozen, gradients update the training shares through the merged proxy under budget and capacity constraints. Periodic evaluations select the lowest-loss mixture. Its quotas specify how many examples to sample from each cluster for training from the original base.

2 METHOD

Framework overview. CLIVO learns target-specific training shares through three decisions: *how to group the data*, *how to evaluate a mixture*, and *how to update its shares* (Figure 1 and Algorithm 1). We define the task (Section 2.1), then present semantic units (Section 2.2), validation learning (Section 2.3), and gradient search (Section 2.4).

2.1 PROBLEM SETUP

Given an instruction pool $\mathcal{D}$, an initial VLM θ_0 , a target capability d , and a budget N , we seek a size- N subset maximizing the target score after instruction tuning. For a partition $\{\mathcal{D}_k\}_{k=1}^K$, we optimize admissible training shares $\mathbf{w} = (w_1, \dots, w_K) \in \mathcal{W}$, with $w_k \geq 0$ and $\sum_k w_k = 1$. These shares determine sample counts $n_k(\mathbf{w}) \approx Nw_k$ totaling N . We fix the within-group rule π_U to uniform sampling without replacement, yielding $\mathcal{S}(\mathbf{w}, \pi_U) = \bigcup_k \pi_U(\mathcal{D}_k, n_k(\mathbf{w}))$. Let $\theta_S = \text{Train}(\theta_0, \mathcal{S})$ denote the model trained with a fixed recipe and $R_d(\theta)$ its target score. Our objective is

$$\mathbf{w}_d^* \in \arg \max_{\mathbf{w} \in \mathcal{W}} F_d(\mathbf{w}; \pi_U), \quad F_d(\mathbf{w}; \pi_U) = \mathbb{E}_{\mathcal{S}} [R_d(\theta_S)]. \quad (1)$$

The expectation accounts for score variation across random subsets drawn with the same shares.

2.2 UNIT: SEMANTIC ALLOCATION

Predefined groups rely on source labels or hand-crafted task rules. Semantic clustering instead provides unsupervised allocation units whose granularity is controlled by the number of clusters. We therefore encode instructions with a fixed external multimodal encoder E , independent of the VLM being tuned, and apply K -means (McQueen, 1967) to obtain $\{\mathcal{D}_k\}_{k=1}^K$. Distinct content within a source can receive different shares, while related content across sources can be allocated together. For nested partitions, independent fine-cluster shares retain every coarse allocation and admit additional compositions, expanding the optimization space. The best attainable target score is therefore no lower and can improve when the additional freedom admits a better mixture (Section A.1). This flexibility also creates a higher-dimensional search problem, making candidate-by-candidate evaluation costly.

We address this with a learned validation objective on reusable proxies (Section 2.3) and gradients that update all shares jointly (Section 2.4).

2.3 SIGNAL: LEARNING THE VALIDATION OBJECTIVE

Real-training feedback for validation selection. We train Y probe models from θ_0 using mixtures $\mathbf{w}_i$. Uniform within-cluster sampling fixes the selection rule so that the probes test changes in composition (Section A.2). Their scores $y_i^{(d)}$ supervise validation selection and weighting.

Proxy losses from merged experts. Training a model for every candidate mixture is costly. Following prior work (Li et al., 2026c; Wang et al., 2026), we approximate its losses by merging cluster experts θ_k trained from θ_0 . The candidate’s shares determine how these fixed experts are combined. With $\Delta_k = \theta_k - \theta_0$, constructing a proxy requires no further training:

$$\tilde{\theta}(\mathbf{w}) = \theta_0 + \sum_{k=1}^K w_k \Delta_k = \sum_{k=1}^K w_k \theta_k. \quad (2)$$

We form a diverse validation candidate pool $\mathcal{V} = \{v_j\}_{j=1}^H$ by uniformly sampling equal numbers of examples from every cluster. For the Y real-probe mixtures, we construct proxies and compute each candidate example’s negative loss, $X_{ij} = -\ell(\tilde{\theta}(\mathbf{w}_i); v_j)$, where ℓ is the mean cross-entropy over its answer tokens. We use these losses and real scores to identify target-relevant validation examples.

Downstream-aligned weighted signals. Search favors low-loss mixtures, so the signal should distinguish strong candidates while ranking the full candidate set. For each example, we average two Spearman correlations between negative loss and target score: one over all fitting probes and one over the half with the highest real scores. Since examples greatly outnumber probes, we retain the top m examples as S_d to limit the number of fitted coefficients (Fan & Lv, 2008). To rank mixtures, we fit centered target scores with nonnegative ridge regression (Hoerl & Kennard, 1970):

$$\hat{\beta}_d = \arg \min_{\beta \geq 0} \left\| \tilde{X}_{\mathcal{I}, S_d} \beta - (\mathbf{y}_{\mathcal{I}}^{(d)} - \bar{y}_d \mathbf{1}) \right\|_2^2 + \lambda \|\beta\|_2^2. \quad (3)$$

Here $\mathcal{I}$ indexes all fitting probes, $\tilde{X}$ contains their standardized negative losses, $\bar{y}_d$ is their mean target score, and λ regularizes the coefficients. Nonnegative coefficients make lower example losses contribute to higher predicted scores. Rescaling the coefficients to raw-loss units and normalizing them gives weights $\alpha_{d,j} \geq 0$ summing to one and the objective

$$J_d(\mathbf{w}) = \mathcal{L}_d(\tilde{\theta}(\mathbf{w})) = \sum_{j \in S_d} \alpha_{d,j} \ell(\tilde{\theta}(\mathbf{w}); v_j). \quad (4)$$

Individual weights preserve differences in how examples’ losses respond to a mixture, which cluster averaging would remove before fitting. Lower J_d corresponds to a higher predicted score, so the conversion preserves the fitted ranking. The validation weights α_d determine how mixtures are evaluated; the training shares $\mathbf{w}$ determine their composition. Detailed settings and the weight conversion are given in Section A.3.

Algorithm 1 CLIVO: target-specific instruction mixing

Require: $\mathcal{D}, \theta_0, E, R_d, N; K, m, T, \{\mathbf{w}_i\}_{i=1}^Y$

Ensure: Shares $\hat{\mathbf{w}}_d$; subset S_d^*

1. Unit: semantic allocation

Encode $\mathcal{D}$ with fixed external encoder E .

Apply K -means; obtain clusters $\{\mathcal{D}_k\}$.

2. Signal: learned validation objective

Train probes at $\{\mathbf{w}_i\}$; evaluate $\mathbf{y}^{(d)}$.

Train cluster experts $\{\theta_k\}$ from θ_0 .

$\mathcal{V} \leftarrow \text{BalancedSample}(\{\mathcal{D}_k\})$

$X_{ij} \leftarrow -\ell(\tilde{\theta}(\mathbf{w}_i); v_j)$

$S_d \leftarrow \text{Top}_m((\rho_{\text{all},j} + \rho_{\text{upper half},j})/2)$

Fit Eq. (3); rescale and normalize to α_d .

3. Learning: mixture search

Freeze $\{\theta_k\}, \alpha_d$.

$\mathbf{w}^0 \leftarrow \arg \min_{\mathbf{w}_i \in \mathcal{W}} J_d(\mathbf{w}_i); \hat{\mathbf{w}}_d \leftarrow \mathbf{w}^0$

for $t = 1, \dots, T$ **do**

$\mathbf{g}^t \leftarrow \nabla_{\mathbf{w}} J_d(\mathbf{w}^{t-1})$ by Eq. (5)

$\mathbf{w}^t \leftarrow \Pi_{\mathcal{W}}(\text{Adam}(\mathbf{w}^{t-1}, \mathbf{g}^t))$

Update best $\hat{\mathbf{w}}_d$ at checkpoints.

end for

Convert $\hat{\mathbf{w}}_d$ to sample counts $\mathbf{n}$ totaling N .

$K S_d^* \leftarrow \bigcup_k \pi_{\cup K}(\mathcal{D}_k, n_k)$

2.4 LEARNING: GRADIENT-BASED MIXTURE SEARCH

The differentiable objective above enables direct, efficient gradient search over the mixture shares. We freeze the experts θ_k and validation weights α_d , and initialize $\mathbf{w}^0$ with the lowest-loss previously evaluated feasible mixture to accelerate convergence. At each step, all positive-weight validation examples contribute to the merged proxy’s loss. Applying the chain rule (Baydin et al., 2018) gives

$$\frac{\partial J_d}{\partial w_k} = \left\langle \nabla_{\theta} \mathcal{L}_d(\theta) \Big|_{\theta=\tilde{\theta}(\mathbf{w})}, \Delta_k \right\rangle. \quad (5)$$

Taking inner products of the accumulated parameter gradient with all K expert differences gives the gradients of every share. Adam updates the shares jointly while the experts remain fixed, so no separate backward pass is needed for each coordinate. Periodic evaluations of the full objective select the lowest-loss iterate $\hat{\mathbf{w}}_d$ among the evaluated checkpoints, including the start.

Finally, we convert the shares into integer counts totaling N , sample uniformly within clusters, and train from θ_0 on $\mathcal{S}(\hat{\mathbf{w}}_d, \pi_U)$. Section B.3 gives search and sampling details.

3 EXPERIMENTS

In this section, we describe the experimental setup (Section 3.1), compare fixed-budget selection quality (Section 3.2), and conduct ablation studies of CLIVO’s components (Section 3.3).

3.1 EXPERIMENTAL SETUP

Model and evaluation. We build a 7B VLM with a Qwen2.5-VL-style architecture (Bai et al., 2025b), following the multimodal training approach of LLaVA-1.5 (Liu et al., 2024a). Alignment and mid-training produce the common base θ_0 for all experts, probes, and final models; construction details are in Section B.1. Evaluation covers General visual understanding (G: MMBench-EN, MMStar, GQA), Parsing (P: OCRBench, ChartQA), and Knowledge and text-centric reasoning (K: MMLU, MMLU-Pro). Each domain’s benchmark mean defines a target; Overall (O) averages all seven benchmarks. We learn one mixture per target, and Average summarizes their four scores. Evaluation protocols are in Section B.3.

Data pool. We collect approximately 40M instructions from public datasets and their derivatives, covering diverse visual content, tasks, and response formats. To keep selection costs manageable, experiments use an 8M candidate pool. All selection methods draw 100K instructions from this pool; Full uses all 8M. Section B.2 provides data statistics.

Baselines. For *sample-level selection*, we choose PPL (Jelinek et al., 1977), CLIP-Score (Hessel et al., 2021), MM-Selector (Ma et al., 2025), PRISM (Bi et al., 2025), ICONS (Wu et al., 2024), DSIR (Xie et al., 2023b), and DataTailor (Yu et al., 2025). For *group-level selection*, we choose COINCIDE (Lee et al., 2024) and SemDeDup (Abbas et al., 2021). We also evaluate two controlled regression variants following mixture-response search (Liu et al., 2024b; Li et al., 2026c): **Acc + Reg.** predicts target accuracy from semantic-cluster shares, whereas **Source + Reg.** searches source-defined shares using a proxy-loss predictor. Random samples 100K examples uniformly; Full uses all 8M. Because ICONS and DSIR require validation or reference data, both receive CLIVO’s selected validation examples for a fair comparison with common reference content. CLIVO samples randomly within clusters by default. To test compatibility with sample ranking, CLIVO + ICONS applies the strong ICONS selector within unchanged quotas. Random, Acc + Reg., Source + Reg., and CLIVO report three-run means; others use one run (Sections B.4 and B.5).

Implementation details. We embed instructions with Qwen3-VL-Embedding-8B (Li et al., 2026b), use $K = 100$ clusters, and train experts with the nominal step-10 budget. Validation learning uses $Y = 128$ real probes and $H = 10,000$ candidates sampled equally across clusters, retaining $m = 64$ examples per target and fitting their weights with $\lambda = 10$. We run 120 gradient updates, retain the best periodically evaluated mixture, and train its 100K-example subset for 100 optimizer steps. Budget definitions, optimizer settings, and evaluation protocols are in Section B.

3.2 TARGET-SPECIFIC SELECTION QUALITY

CLIVO achieves competitive fixed-budget selection.

With random within-cluster sampling, CLIVO reaches a four-target mean of 69.20, exceeding the strongest baseline, Acc + Reg., by 1.16 points and the strongest sample-level selector, ICONS, by 1.55 points (Table 1). It also surpasses Source + Reg. by 1.99 points. CLIVO leads the baselines on Overall, Parsing, and Knowledge with scores of 68.13, 79.64, and 61.76, showing the value of target-specific semantic allocation without ranking individual training examples. Baselines favor different capabilities: CLIP-Score, which measures image-text alignment, achieves the highest baseline Parsing score (79.42); DSIR, which matches a reference distribution, achieves the highest baseline General score (67.56). ICONS and DSIR also obtain competitive results using CLIVO’s selected validation examples, showing that this reference data can support existing selectors. Benchmark-level results are in Section B.6.

Table 1: **Selection quality at a 100K budget.** O/G/P/K: four targets; Average: their mean. Random, Acc + Reg., Source + Reg., and CLIVO average three runs. Bold/underline: best/second-best at 100K.

Method	O	G	P	K	Average
Full (8M)	71.01	69.23	80.87	63.81	71.23
Random	66.61	63.57	77.73	60.05	66.99
PPL	64.26	60.78	75.65	58.09	64.70
CLIP-Score	66.16	62.23	79.42	58.80	66.65
MM-Selector	66.46	64.05	76.96	59.59	66.77
PRISM	67.27	65.73	76.97	59.86	67.46
ICONS	67.32	64.12	79.30	59.88	67.65
DSIR	67.31	<u>67.56</u>	72.62	60.42	66.98
DataTailor	65.72	61.50	77.58	60.20	66.25
SemDeDup	64.00	57.53	77.87	59.84	64.81
COINCIDE	66.20	62.65	78.14	59.60	66.65
Acc + Reg.	67.53	65.44	77.80	61.40	68.04
Source + Reg.	67.00	62.91	79.12	59.80	67.21
CLIVO	68.13	67.28	<u>79.64</u>	<u>61.76</u>	<u>69.20</u>
CLIVO + ICONS	<u>68.05</u>	68.27	79.96	62.09	69.59

Allocation and ranking are complementary. Keeping CLIVO’s quotas fixed and replacing random sampling with within-cluster ICONS improves most targets, raising the four-target mean from 69.20 to 69.59 and General by 0.99 points. The combination also exceeds global ICONS on all four targets, with gains of 4.15 points on General and 1.94 on the four-target mean. Thus, the same ranking method is more effective when it fills target-specific quotas: allocation determines how much of each content type to include, and ranking chooses its representatives. Global sample ranking alone leaves these composition choices implicit (Section C.2).

3.3 ABLATION STUDIES

We examine three questions: whether finer allocation improves selection, whether proxy losses provide reliable target feedback, and whether gradient search uses that feedback to improve mixtures.

The value of finer allocation depends on the target. We isolate allocation freedom by tying the same 100 expert coefficients into 20 or 50 parent shares, or optimizing all 100 independently. The proxy bank, validation objective, starting composition, and training budgets stay fixed. Tying is algebraically equivalent to premerging child experts at fixed ratios (Section B.7).

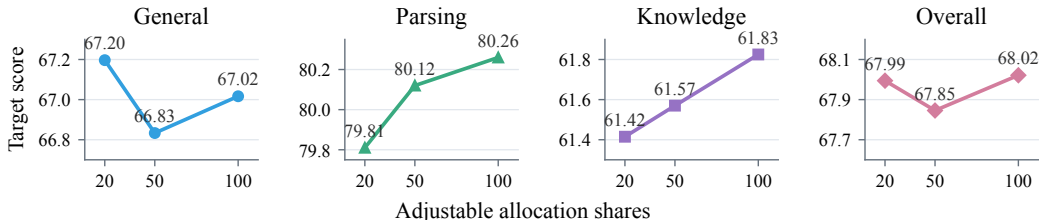


Figure 2: **Allocation freedom and downstream performance.** Each panel varies adjustable shares on the same 100-expert bank. Scores use 100K examples, random within-cluster sampling, and one training seed. All vertical axes span 0.6 points. Search records: Table 10.

In Figure 2, 100 adjustable shares give the highest scores on most targets. Parsing and Knowledge improve by 0.45 and 0.41 points over 20 shares, whereas General performs best at 20. Finer allocation is useful when a target benefits from separating content within coarse groups; the preferred granularity therefore depends on the target.

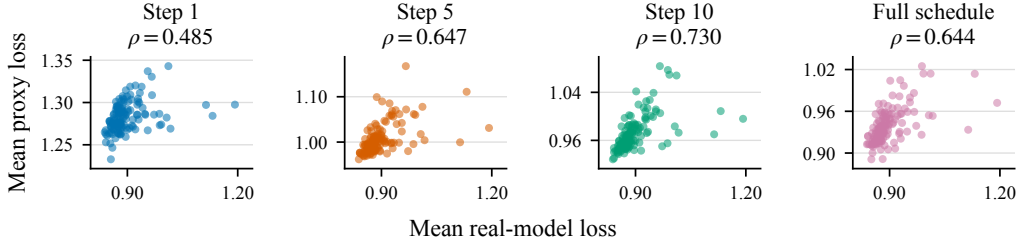


Figure 3: **Expert budget and proxy-real loss agreement.** Each point compares mean proxy and real-model losses for the same mixture on a shared 10,000-example panel. Panels show different expert budgets; ρ is Spearman correlation across 128 mixtures. Budget details: Section B.8.

Moderate-budget experts capture real-model loss variation. To test proxy reliability, we compare expert checkpoints indexed by $b \in \{1, 5, 10\}$ and a full-training reference on identical mixtures and examples (Figure 3). Proxy-real loss correlation increases from 0.485 to 0.730 across the short budgets, with checkpoint 10 giving the strongest agreement. Full training lowers absolute loss but does not improve this agreement. Checkpoint 10 also improves held-out downstream-score prediction over checkpoint 1 on every target (Table 11). These results support its use as the shared proxy bank; budget definitions and protocols are in Section B.8.

Individual responses preserve useful validation information. We compare cluster-level and sample-level aggregation, each with equal or learned weights, on the same 64 examples per fitting fold (Table 2). Cluster-level rules average losses within each represented cluster before weighting, while sample-level rules retain each example’s response to mixture changes. Learned sample weights give the highest held-out Spearman on all four targets, improving over equal sample averaging by up to 0.121. Their advantage over weighted cluster means supports retaining individual responses: fitting can emphasize informative examples whose signals would otherwise be diluted by averaging. The fit concentrates weight on a subset of examples (Section A.3), emphasizing their contribution to target evaluation.

Table 2: **Validation aggregation.** Held-out Spearman over 20 folds, with 64 matched examples. C/S: cluster/sample; E/W: equal/weighted. Maximum correlations are bold.

Target	C-E	C-W	S-E	S-W
General	0.621	0.684	0.619	0.714
Parsing	0.397	0.430	0.379	0.461
Knowledge	0.586	0.594	0.589	0.596
Overall	0.469	0.569	0.470	0.591

Learned signals guide mixture search. Held-out correlations show that the learned objectives predict downstream rankings for all four targets (Table 3). Gradient search lowers every fixed objective, reducing loss by 0.056–3.724% from the evaluated starts. These signals therefore support direct optimization of the mixture shares. All four final-training means also exceed the best historical probe scores, by 0.21–0.68 points. Section B.9 provides search trajectories and training details.

Table 3: **From feedback to final training.** $K = 100$. CV- ρ : held-out Spearman; t^* : best step; D_J : loss decrease (%); B : best probe; $\bar{F}$: three-run mean; $\Delta F = \bar{F} - B$. Protocols: Section B.9.

Target	CV- ρ	t^*	D_J	B	$\bar{F}$	ΔF
Gen.	0.714	100	3.724	66.600	67.28	+0.68
Pars.	0.461	120	0.293	79.310	79.64	+0.33
Know.	0.596	40	0.056	61.465	61.76	+0.30
Ovrl.	0.591	100	2.040	67.919	68.13	+0.21

4 INTERPRETABLE, FLEXIBLE, AND REUSABLE DATA CURATION

In this section, we show how CLIVO supports **interpretable, flexible, and reusable data curation**. Semantic visualizations make target-specific selection decisions inspectable (Section 4.1); modular components accommodate different applications, while shared experts and feedback support reuse across targets and allocation granularities (Section 4.2).

4.1 DATA VISUALIZATION AND MANAGEMENT

Targets favor different, concentrated mixtures. We use GPT-based task annotations to group clusters into coarse capability categories for interpretation (Section C.1). Figure 4 shows distinct

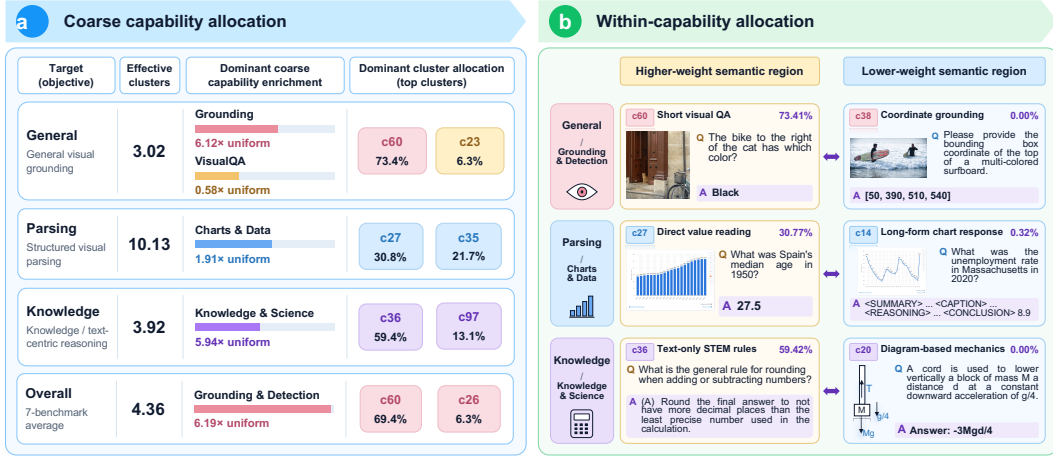


Figure 4: **Target-specific composition, from distributions to examples.** (a) Allocation concentration and dominant semantic regions. (b) Paired clusters within the same coarse capability receive different training shares. Percentages give each cluster’s share of the complete training budget. Uniform allocates 1% to each of the 100 clusters; × uniform compares a category’s total learned share with its share under this equal-cluster allocation. Definitions and cases: Section C.1.

preferences: General and Overall emphasize c60, illustrated by short visual questions; Parsing favors c27, illustrated by chart reading; and Knowledge favors c36, illustrated by text-based STEM supervision. Allocation is concentrated but target-dependent: the entropy-based effective cluster count is 10.13 for Parsing and 3.02–4.36 for the other targets, out of 100 clusters. Labels and examples help interpret these differences (Table 15).

Within-category contrasts highlight the value of fine-grained allocation. Clusters within the same coarse capability receive markedly different shares (Figure 4(b)). Within *Grounding & Detection*, General assigns 73.41% to c60, illustrated by short visual questions, and zero to c38, illustrated by bounding-box coordinates. Within *Charts & Data*, Parsing assigns 30.77% to c27, illustrated by direct value reading, and 0.32% to c14, illustrated by long responses. Within *Knowledge & Science*, Knowledge assigns 59.42% to c36, illustrated by text-only STEM tasks, and zero to c20, illustrated by mechanics diagrams. These contrasts reveal target preferences within broad categories, helping explain why independently adjustable fine-grained shares can be useful (Figure 2).

4.2 FLEXIBLE AND REUSABLE DATA CURATION

Modular components enable flexible selection. CLIVO separates semantic organization, target evaluation, and within-cluster selection, so each component can be configured for the application. Its quotas also integrate with state-of-the-art sample-level selectors: allocation determines how much of each content type to include, and ranking chooses the examples within those quotas. The CLIVO + ICONS results demonstrate this combination (Section 3.2), allowing existing ranking methods to benefit from target-specific composition.

One-time construction supports new targets. After constructing the embeddings, semantic partition, expert bank, real probe models, and candidate-loss cache, CLIVO can reuse these assets for new targets on the same base and partition. Evaluating the saved probes on a new benchmark provides the target scores; validation-example selection and weighting then operate on cached losses, followed by gradient search under the new objective. Thus, target adaptation requires neither retraining experts or probes nor recomputing the cached per-example losses (Section C.3).

Shared gradients support finer allocation. As allocation becomes finer on the same expert bank, a single backward pass supplies gradients to update all mixture weights jointly (Eq. (5)). CLIVO thus avoids fitting an additional regression model over the higher-dimensional mixture space. In the $K = 20/50/100$ experiments, costs to reach the best recorded losses show no consistent monotonic increase across targets as allocation becomes finer (Table 10). With the same expert bank and

validation objectives, the four gradient searches use 26.6% fewer peak-BF16-normalized GPU-hours than labeling 4,096 mixtures for regression, a budget at which held-out proxy-loss prediction shows diminishing gains. This reuse of model gradients supports fine-grained search without an additional mixture-response fitting stage (Sections C.3 and C.4).

5 RELATED WORK

Sample-level methods. Sample-level methods score individual examples and select a subset according to those scores. Their signals fall into four broad categories. *Quality and alignment* scores assess instruction quality and complexity in DEITA (Liu et al., 2023b), human preferences in Align2LLaVA (Huang et al., 2025), and image-text compatibility in CLIP-Score (Hessel et al., 2021). *Model-response* signals capture necessity in MLLM-Selector and selector confidence in Once-For-All (Dong et al., 2026). *Influence-based* methods estimate training effects: TIVE combines instance influence with task difficulty, ICONS seeks cross-task consensus, and BIDS (Dai et al., 2025) balances task-dependent influence scales in text instruction tuning. *Distribution and diversity* signals address coverage: DSIR (Xie et al., 2023b) resamples toward a reference distribution, PRISM (Bi et al., 2025) uses visual-feature geometry, DataTailor (Yu et al., 2025) combines informativeness, uniqueness, and representativeness, and PreSel (Safaei et al., 2025) clusters images before generating instructions. CoIDO jointly models importance and diversity. These categories can overlap with allocation; TIVE and DataTailor also determine task quotas. CLIVO learns target-specific semantic quotas that can be filled using existing sample-level selectors.

Group-level methods. Group-level methods partition a pool and determine how much training data to draw from each group. They can be classified by how shares are chosen. *Heuristic allocation* converts group statistics into quotas, as in COINCIDE’s use of concept-skill density and transferability. *Online adaptive weighting* updates shares during training; DoReMi uses group distributionally robust optimization on reference-relative losses. *Response-model-based search* fits mixture outcomes, using flexible regressors in RegMix and CLIMB or parametric laws in Data Mixing Laws (Ye et al., 2024), BiMix (Ge et al., 2024), and CAMEL. The latter additionally links validation losses to benchmark scores. Expert reuse changes how these response models obtain feedback: MDE (Belenki et al., 2025) constructs features from combined expert predictions, while DeMix and MergeMix evaluate parameter-merged experts to train LightGBM predictors (Ke et al., 2017). *Direct proxy evaluation* compares merged candidates without a fitted mixture-response function (Berasi et al., 2026). Related work on adaptive merging optimizes coefficients using entropy minimization to improve the merged model itself (Yang et al., 2024). CLIVO learns a sample-loss objective and differentiates it through fixed experts to optimize training shares.

6 CONCLUSION

We presented CLIVO, a target-aware instruction-selection framework connecting fine-grained semantic allocation, learned validation objectives, and differentiable mixture search. Learning from individual validation losses turns real-training feedback into an objective for optimizing data composition. Experiments show that learned quotas improve fixed-budget selection and complement sample ranking. This distinction between allocating content and ranking examples provides a practical way to adapt a shared instruction pool to different capability needs. The learned objective links these allocation decisions to target performance, while semantic visualization reveals which content is emphasized. Modular components accommodate different encoders and selectors, and shared experts support reuse across targets and granularities. Together, these properties make the selected subset both adaptable and open to inspection. Future work will explore adaptive probe selection to reduce real-training feedback, transfer of learned validation objectives across base models and training budgets, and joint optimization of multiple capability targets. Hierarchical semantic units could also adapt allocation granularity to the task as instruction pools grow, extending the framework to evolving data collections. We hope this work encourages joint study of example quality and data composition for transparent, target-aware curation.

AI USE STATEMENT

AI tools were used to polish the manuscript’s language and improve readability. GPT also assisted semantic annotation for data visualization; the annotation procedure and human verification are described in Section C.1. The authors independently developed the core research ideas and methodology, designed and conducted the experiments, and formulated the scientific conclusions. The authors take full responsibility for this work.

ETHICS STATEMENT

This work uses publicly available datasets and derived instruction data in accordance with their applicable licenses and terms of use. We document the data sources and processing scope in Section B.2. Our experiments do not involve recruiting human participants or collecting new personal information. Data processing and reporting follow applicable privacy requirements, and qualitative examples are presented without personally identifying information. CLIVO is intended to support transparent, target-aware data selection for research.

REPRODUCIBILITY STATEMENT

Section 2 specifies validation-example screening, nonnegative ridge fitting, weight conversion, expert merging, and mixture search. Section A gives the method derivations and validation objective; Section B documents data, training, baselines, and detailed results. Annotation procedures, example provenance, and reuse accounting appear in Section C.

REFERENCES

- Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication, 2023. URL <https://arxiv.org/abs/2303.09540>, 2021.
- Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025.
- Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.
- Shuai Bai, Ke qin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *ArXiv*, abs/2502.13923, 2025b. URL <https://api.semanticscholar.org/CorpusID:276449796>.
- Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, 18 (153):1–43, 2018.
- Lior Belenki, Alekh Agarwal, Tianze Shi, and Kristina Toutanova. Optimizing pre-training data mixtures with mixtures of data expert models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32570–32587, 2025.
- Davide Berasi, Matteo Farina, Massimiliano Mancini, and Elisa Ricci. Linear model merging unlocks simple and scalable multimodal data mixture optimization. *arXiv preprint arXiv:2602.04937*, 2026.
- Jinhe Bi, Yifan Wang, Danqi Yan, Wenke Huang, Zengjie Jin, Xiaowen Ma, Sikuan Yan, Artur Hecker, Mang Ye, Xun Xiao, et al. Prism: Self-pruning intrinsic selection method for training-free multimodal data selection. *arXiv preprint arXiv:2502.12119*, 2025.

- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.
- 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.
- Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.
- Shizhe Diao, Yu Yang, Yonggan Fu, Xin Dong, Dan Su, Markus Kliegl, Zijia Chen, Peter Belcak, Yoshi Suhara, Hongxu Yin, et al. Climb: Clustering-based iterative data mixture bootstrapping for language model pre-training. *arXiv e-prints*, pp. arXiv–2504, 2025.
- Mingkang Dong, Hongyi Cai, Xiwen Lei, Jie Li, Tao Zhang, and Muxin Pu. Once-for-all: A train-once and select-anytime framework for multimodal instruction tuning. *arXiv preprint arXiv:2605.26761*, 2026.
- Yifan Du, Hangyu Guo, Kun Zhou, Wayne Xin Zhao, Jinpeng Wang, Chuyuan Wang, Mingchen Cai, Ruihua Song, and Ji-Rong Wen. What makes for good visual instructions? synthesizing complex visual reasoning instructions for visual instruction tuning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 8197–8214, 2025.
- Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 70(5):849–911, 2008.
- Ce Ge, Zhijian Ma, Daoyuan Chen, Yaliang Li, and Bolin Ding. Bimix: A bivariate data mixing law for language model pretraining. *arXiv preprint arXiv:2405.14908*, 2024.
- Yunhao Gou, Hansi Yang, Zhili Liu, Kai Chen, Yihan Zeng, Lanqing Hong, Zhenguo Li, Qun Liu, Bo Han, James Kwok, et al. Corrupted but not broken: Understanding and mitigating the negative impacts of corrupted data in visual instruction tuning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25945–25971, 2025.
- Hongyi He, Xiao Liu, Zhenghao Lin, Mingni Tang, Yi Cheng, Jintao Wang, Wenjie Li, Peng Cheng, and Yeyun Gong. Learning from the best, differently: A diversity-driven rethinking on data selection. *arXiv preprint arXiv:2510.18909*, 2025.
- Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.
- Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
- Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.
- Ailin Huang, Chengyuan Yao, Chunrui Han, Fanqi Wan, Hangyu Guo, Haoran Lv, Hongyu Zhou, Jia Wang, Jian Zhou, Jianjian Sun, et al. Step3-vl-10b technical report. *arXiv preprint arXiv:2601.09668*, 2026.
- Hongzhe Huang, Jiang Liu, Zhewen Yu, Li Cai, Dian Jiao, Wenqiao Zhang, Siliang Tang, Juncheng Li, Hao Jiang, Haoyuan Li, et al. Align2llava: Cascaded human and large language model preference alignment for multi-modal instruction curation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 8759–8781, 2025.

- Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. Perplexity—a measure of the difficulty of speech recognition tasks. *The journal of the Acoustical Society of America*, 62(S1): S63–S63, 1977.
- Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.
- Jaewoo Lee, Boyang Li, and Sung Ju Hwang. Concept-skill transferability-based data selection for large vision-language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5060–5080, 2024.
- Jingwei Li, Xinran Gu, and Jingzhao Zhang. Capacity-aware mixture law enables efficient llm data optimization. *arXiv preprint arXiv:2603.08022*, 2026a.
- Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibong Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, et al. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026b.
- Shengrui Li, Fei Zhao, Kaiyan Zhao, Jieying Ye, Haifeng Liu, Fangcheng Shi, Zheyong Xie, Yao Hu, and Shaosheng Cao. Decouple searching from training: Scaling data mixing via model merging for large language model pre-training. *arXiv preprint arXiv:2602.00747*, 2026c.
- Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26679–26689. IEEE, 2024.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023a.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296. IEEE, 2024a.
- 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. *arXiv preprint arXiv:2407.01492*, 2024b.
- Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *arXiv preprint arXiv:2312.15685*, 2023b.
- Zikang Liu, Kun Zhou, Wayne Xin Zhao, Dawei Gao, Yaliang Li, and Ji-Rong Wen. Less is more: High-value data selection for visual instruction tuning. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 3712–3721, 2025.
- Run Luo, Haonan Zhang, Longze Chen, Ting-En Lin, Xiong Liu, Yuchuan Wu, Min Yang, Yongbin Li, Minzheng Wang, Pengpeng Zeng, et al. Mmevol: Empowering multimodal large language models with evol-instruct. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19655–19682, 2025.
- Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Jiayi Ji, Jie Lou, Debing Zhang, and Rongrong Ji. Mllm-selector: Necessity and diversity-driven high-value data selection for enhanced visual instruction tuning. *arXiv preprint arXiv:2503.20502*, 2025.
- James B McQueen. Some methods of classification and analysis of multivariate observations. In *Proc. of 5th Berkeley Symposium on Math. Stat. and Prob.*, pp. 281–297, 1967.
- Shruthan Radhakrishna, Aman Tiwari, Aanjaneya Shukla, Masoud Hashemi, Rishabh Maheshwary, Shiva Krishna Reddy Malay, Jash Mehta, Pulkit Pattnaik, Saloni Mittal, Khalil Slimi, et al. Apriel-1.5-15b-thinker. *arXiv preprint arXiv:2510.01141*, 2025.

- Bardia Safaei, Faizan Siddiqui, Jiacong Xu, Vishal M Patel, and Shao-Yuan Lo. Filter images first, generate instructions later: Pre-instruction data selection for visual instruction tuning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14247–14256. IEEE, 2025.
- James E Smith and Robert L Winkler. The optimizer’s curse: Skepticism and postdecision surprise in decision analysis. *Management Science*, 52(3):311–322, 2006.
- Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, et al. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*, 2025a.
- Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.
- Meituan LongCat Team, Bairui Wang, Bin Xiao, Bo Zhang, Bolin Rong, Borun Chen, Chang Wan, Chao Zhang, Chen Huang, Chen Chen, et al. Longcat-flash-omni technical report. *arXiv preprint arXiv:2511.00279*, 2025b.
- Jiapeng Wang, Changxin Tian, Kunlong Chen, Ziqi Liu, Jiaxin Mao, Wayne Xin Zhao, Zhiqiang Zhang, and Jun Zhou. Mergemix: Optimizing mid-training data mixtures via learnable model merging. *arXiv preprint arXiv:2601.17858*, 2026.
- Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.
- Xindi Wu, Mengzhou Xia, Rulin Shao, Zhiwei Deng, Pang Wei Koh, and Olga Russakovsky. Icons: Influence consensus for vision-language data selection. *arXiv preprint arXiv:2501.00654*, 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, 2023a.
- Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36:34201–34227, 2023b.
- Zhiyang Xu, Chao Feng, Rulin Shao, Trevor Ashby, Ying Shen, Di Jin, Yu Cheng, Qifan Wang, and Lifu Huang. Vision-flan: Scaling human-labeled tasks in visual instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 15271–15342, 2024.
- Yichen Yan, Ming Zhong, Qi Zhu, Xiaoling Gu, Jinpeng Chen, and Huan Li. Coido: Efficient data selection for visual instruction tuning via coupled importance-diversity optimization. *Advances in Neural Information Processing Systems*, 38:167045–167073, 2026.
- Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations*, volume 2024, pp. 22743–22763, 2024.
- Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. *arXiv preprint arXiv:2403.16952*, 2024.
- Weijie Yin, Yongjie Ye, Fangxun Shu, Yue Liao, Zijian Kang, Hongyuan Dong, Haiyang Yu, Dingkan Yang, Jiacong Wang, Han Wang, et al. Sail-vl2 technical report. *arXiv preprint arXiv:2509.14033*, 2025.
- Qifan Yu, Zhebei Shen, Zhongqi Yue, Yang Wu, Bosheng Qin, Wenqiao Zhang, Yunfei Li, Juncheng Li, Siliang Tang, and Yueting Zhuang. Mastering collaborative multi-modal data selection: A focus on informativeness, uniqueness, and representativeness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 155–165, 2025.

Haotian Zhang, Mingfei Gao, Zhe Gan, Philipp Dufter, Nina Wenzel, Forrest Huang, Dhruti Shah, Xianzhi Du, Bowen Zhang, Yanghao Li, et al. Mm1. 5: Methods, analysis & insights from multimodal llm fine-tuning. In *International Conference on Learning Representations*, volume 2025, pp. 64669–64714, 2025.

APPENDIX CONTENTS

A	Method Details	15
A.1	Finer Allocation Can Improve the Attainable Optimum	15
A.2	Mixture Interpretation of Within-Cluster Sampling	15
A.3	Learning the Validation Objective	15
A.4	Direct Search Avoids an Additional Response-Fitting Error	16
B	Experimental Details and Additional Results	16
B.1	Model Construction	16
B.2	Data Pool	16
B.3	Training and Evaluation Settings	17
B.4	Baseline Selection Methods	18
B.5	Repeated Runs	18
B.6	Benchmark-Level Results	19
B.7	Allocation Freedom on a Fixed Expert Bank	19
B.8	Expert Training Budget and Proxy Agreement	20
B.9	Gradient Search Progress	21
C	Data Interpretation, Flexibility, and Reuse	21
C.1	Data Annotation and Composition	21
C.2	Why Allocation and Sample Ranking Complement Each Other	24
C.3	Reusing Feedback Across Targets and Granularities	24
C.4	Incremental Search Budget	25

A METHOD DETAILS

A.1 FINER ALLOCATION CAN IMPROVE THE ATTAINABLE OPTIMUM

A coarse allocation fixes the proportions of fine clusters within each parent. Write this restriction as $\mathbf{w} = B\mathbf{z}$, where each column of B contains the nonnegative child proportions of one parent and sums to one. The fine family $\mathcal{W}$ and its coarse restriction $\mathcal{Z} = \{\mathbf{z} : B\mathbf{z} \in \mathcal{W}\}$ satisfy

$$\sup_{\mathbf{z} \in \mathcal{Z}} F_d(B\mathbf{z}) \leq \sup_{\mathbf{w} \in \mathcal{W}} F_d(\mathbf{w}). \quad (6)$$

Every coarse mixture is also a fine mixture, so optimizing over the larger family cannot lower the best attainable score. The improvement can be strict when a better composition requires changing the fixed within-parent proportions. This is the benefit of fine-grained allocation: it preserves coarse choices and allows additional ones.

A.2 MIXTURE INTERPRETATION OF WITHIN-CLUSTER SAMPLING

With uniform within-cluster sampling fixed, composition is the selection decision being varied. For cluster size M_k , draw n_k examples uniformly without replacement and let $r_k = n_k/N$ be its realized share. For any fixed per-example quantity f ,

$$\mathbb{E} \left[\frac{1}{N} \sum_{x \in \mathcal{S}} f(x) \right] = \sum_k r_k \left(\frac{1}{M_k} \sum_{x \in \mathcal{D}_k} f(x) \right). \quad (7)$$

Indeed, each example in cluster k is included with probability n_k/M_k ; summing its expected contribution gives the identity. Thus the subset represents the specified mixture in expectation, without introducing a preference for particular examples within a cluster. Under the fixed training recipe, a real probe supplies a noisy observation of that mixture’s expected downstream score. Repeated sampling reduces variation in the mean under the same within-cluster rule.

A.3 LEARNING THE VALIDATION OBJECTIVE

The candidate panel contains 100 examples from each of 100 clusters. We evaluate proxies at the same 128 mixtures as the real probes. Each loss averages cross-entropy over that example’s answer tokens, so response length is normalized before examples are aggregated.

Screening and fitting. Let $X_{ij} = -\ell(\tilde{\theta}(\mathbf{w}_i); v_j)$. On fitting indices $\mathcal{I}$, the screening score is

$$q_{d,j} = \frac{1}{2}\rho_{\text{Sp}}(X_{\mathcal{I},j}, \mathbf{y}_{\mathcal{I}}^{(d)}) + \frac{1}{2}\rho_{\text{Sp}}(X_{\mathcal{I}_d^+,j}, \mathbf{y}_{\mathcal{I}_d^+}^{(d)}), \quad (8)$$

where $\mathcal{I}_d^+$ is the higher-scoring half of the fitting probes. We retain the top $m = 64$ examples and standardize their negative losses using fitting-probe means μ_j and standard deviations s_j . We then fit their coefficients by Eq. (3), with $\lambda = 10$. The fit uses centered numerical target scores; Spearman correlation measures the resulting ordering.

Returning to loss units. With $c_j = \hat{\beta}_j/s_j$, $C_d = \sum_j c_j > 0$, and $\alpha_{d,j} = c_j/C_d$, the fitted score is

$$\hat{y}_i = \bar{y}_d + \sum_j \hat{\beta}_j \frac{X_{ij} - \mu_j}{s_j} = b_d - C_d J_d(\mathbf{w}_i), \quad b_d = \bar{y}_d - \sum_j c_j \mu_j. \quad (9)$$

Consequently, minimizing the normalized weighted loss preserves the fitted score ranking. The final objective is fitted on all 128 probes and frozen for search.

Preserving individual responses. A weighted average of cluster-mean losses assigns the same effective weight to every selected example within a cluster. Individual weights include this special case but also distinguish examples with different responses to the training mixture. They therefore preserve information that a cluster mean removes before fitting.

Matched evaluation. The four aggregation rules in Table 2 share the same 64 screened examples in each of five repeated four-fold splits. Screening, standardization, and weight fitting use only the 96 fitting probes; the remaining 32 evaluate ranking. Cluster-level rules average losses within each represented cluster; sample-level rules retain individual losses. Equal weighting uses a uniform average; learned weighting uses the same nonnegative ridge fit. The sample-weighted fits retain, on average, 15.3/12.2/13.6/16.1 positive weights for General/Parsing/Knowledge/Overall, respectively. All splits reuse the same 128 development probes.

A.4 DIRECT SEARCH AVOIDS AN ADDITIONAL RESPONSE-FITTING ERROR

Suppose a response regressor h approximates the fixed validation objective J_d on a candidate family $\mathcal{C}$, with $\sup_{\mathcal{C}} |h - J_d| \leq \xi$. If its selected mixture $\hat{\mathbf{w}}_h$ is η -optimal for h , then

$$J_d(\hat{\mathbf{w}}_h) - \min_{\mathcal{C}} J_d \leq \eta + 2\xi. \quad (10)$$

To see this, let $\mathbf{w}_J$ minimize J_d . Then $J_d(\hat{\mathbf{w}}_h) \leq h(\hat{\mathbf{w}}_h) + \xi \leq h(\mathbf{w}_J) + \eta + \xi \leq J_d(\mathbf{w}_J) + \eta + 2\xi$. An η -optimal solution to J_d has a gap of at most η , avoiding the additional 2ξ term from regression.

For the downstream interpretation, assume $|F_d(\mathbf{w}) - (a_d - c_d J_d(\mathbf{w}))| \leq \epsilon_d$ on $\mathcal{C}$, with $c_d > 0$. Direct η -optimal search then gives $\max_{\mathcal{C}} F_d - F_d(\hat{\mathbf{w}}) \leq 2\epsilon_d + c_d \eta$, while the response-fit route gives $2\epsilon_d + c_d(\eta + 2\xi)$. Direct search uses the same signal without the extra regression term.

B EXPERIMENTAL DETAILS AND ADDITIONAL RESULTS

B.1 MODEL CONSTRUCTION

We retain the pretrained Qwen language backbone and randomly initialize the visual head and alignment MLP. Alignment jointly trains the visual head and MLP while keeping the language backbone fixed, using [LLaVA-Pretrain \(558K\)](#). We then perform multimodal mid-training on [LLaVA-OneVision-1.5 Mid-Training-85M \(An et al., 2025\)](#) to obtain the common base θ_0 .

B.2 DATA POOL

Some public datasets combine multiple constituent sources. We manually separate these into smaller source branches to preserve their identities, then remove duplicate and noisy records. The resulting instruction collection contains approximately 40M processed records across 30 branches. Table 4 lists

the upstream releases and processed counts. Upstream sizes refer to the indicated release or branch; processed rows follow our training format, which can combine multiple turns into one conversation. Derived branches can therefore differ in counting unit from their parent releases. Sampling 20% from each semantic region gives the shared 8,007,947-example candidate pool. Every selection method draws its 100K subset from this pool; Full uses the complete candidate pool.

Table 4: **Instruction-pool sources.** Public release sizes and processed source-branch rows before 20% sampling. Derived branches list parent-release sizes.

Dataset or branch	Link	Upstream size	Processed rows
ECD-10K Images	HF	~321.5K	321,544
Honey-Data-1M	HF	1,072,111	1,070,304
Llama-Nemotron-VLM v1, VQA-9	HF	46,745	46,745
MGrounding-630K	HF	~630K	623,979
MM-IFInstruct-23K	HF	22,554	22,555
MM-IFInstruct-23K, second branch	HF	22,554	22,551
SPIQA, training split	HF	262,524	262,524
CMM-Math, image-bearing train	HF	6,272	5,004
DocReason25K	HF	25,877	25,741
MP-DocReason51K	HF	51,754	49,959
Enigmata, generated snapshot	HF	217,541	4,758
IQBench	Code	500	500
LLaVA-OneVision Stage-2, video	HF	~350K videos	~1,090,000
LLaVA-OneVision Stage-2, image	HF	~2.7M	2,633,205
LLaVA-OneVision Stage-2, text	HF	~450K	522,701
LLaVA-OneVision-1.5 Instruct	HF	~21.2M	21,592,501
LLaVAR-2 Hybrid-Instruct	HF	~42K images	43,660
LongAlign-10K	HF	9,888	9,861
LongWriter-6K	HF	6,000	5,926
Magpie-Align, filtered	HF	500K	479,743
MMDU-45K	HF	45,208	28,979
MM-MathInstruct-3M	HF	2,871,988	2,871,988
MTVQA, training split	HF	6,678 images	6,678
SmolTalk2, SFT	HF	3,383,242	3,382,846
MMPR-v1.2	HF	815,293	814,687
Tulu-3 Persona-IF	HF	29,980	29,980
VisualWebInstruct, conversation	HF	1,004,070	1,004,068
VisualWebInstruct, text derivative	HF	1,004,070	558,467
COYO-derived instructions	HF	746,972,269	99,965
WildChat branch	HF	3,199,860	2,398,331

B.3 TRAINING AND EVALUATION SETTINGS

We report compute on a common 64-H100-equivalent basis. Model training uses AdamW, BF16, and zero weight decay, with the schedules in Table 5. Experts use a constant learning rate; other stages use cosine decay with 1% warmup. Final tuning uses a global batch of 128 packed sequences for 100 updates. Training/loader seed pairs are 42/52, 43/53, and 44/54; single-run comparisons use 42/52.

Table 5: **Training schedules.** Expert checkpoint indices are defined in Section B.8.

Stage	Updates / checkpoint	Peak learning rate	Token budget
Alignment	4,500	10^{-3}	16,384
Mid-training	35,999 of 40,000	10^{-5}	8,192
Cluster experts	1 / 5 / 10	10^{-5}	16,384
Real probes	50	10^{-5}	16,384
Final instruction tuning	100	10^{-5}	16,384

Selection settings. Search updates only the FP32 mixture shares using Adam at learning rate 10^{-3} for 120 steps, with zero weight decay. Each update uses all positive-weight validation examples; full BF16 evaluations every 20 steps, including step 0, select the best mixture. Integer quotas total 100K by rounding shares down and distributing the remainder to the largest fractional parts, followed by uniform within-cluster sampling.

Target scores. For target benchmark set $\mathcal{B}_d$, $y_i^{(d)} = |\mathcal{B}_d|^{-1} \sum_{b \in \mathcal{B}_d} s_{i,b}$. Scores use percentage-point units, including OCRBench rescaled to that range. Overall averages all seven benchmarks; Average in the main table averages the four target columns.

Table 6: **Benchmark-to-target mapping.** Domain targets average their constituent benchmark scores equally. Overall averages all seven scores, not the three domain means.

Benchmark	Domain target	Domain weight	Overall weight
MMBench-EN	General	1/3	1/7
MMStar	General	1/3	1/7
GQA	General	1/3	1/7
OCRBench	Parsing	1/2	1/7
ChartQA	Parsing	1/2	1/7
MMLU	Knowledge	1/2	1/7
MMLU-Pro	Knowledge	1/2	1/7

B.4 BASELINE SELECTION METHODS

All selection methods construct 100K-example subsets from the same candidate pool and use the same final-training recipe. **Random** samples uniformly without replacement; **Full** uses all 8M.

Sample-level methods. **PPL** (Jelinek et al., 1977) favors examples with lower response perplexity, using model likelihood as the selection signal. **CLIP-Score** (Hessel et al., 2021) uses image-text embedding similarity to favor visually aligned instructions. **MLLM-Selector** (Ma et al., 2025), labeled MM-Selector in our tables, combines necessity scores from a seed instruction-tuned model with diversity-aware sampling. **PRISM** (Bi et al., 2025) derives training-free selection scores from intrinsic visual features and their redundancy. **ICONS** (Wu et al., 2024) estimates gradient influence on reference examples and aggregates task-level evidence by consensus. **DSIR** (Xie et al., 2023b) estimates source-to-reference importance ratios in a lightweight feature space and resamples toward the reference distribution. **DataTailor** (Yu et al., 2025) combines informativeness, within-cluster uniqueness, and cross-cluster representativeness. Because ICONS and DSIR require validation or reference data, both receive CLIVO’s selected validation examples to keep target-specific reference content consistent across comparisons.

Group-based methods. **SemDeDup** (Abbas et al., 2021) identifies semantically similar examples in embedding clusters and removes redundant representatives. **COINCIDE** (Lee et al., 2024) organizes examples by concept-skill representations and selects using cluster density and transferability. Our two regression controls follow mixture-response modeling (Liu et al., 2024b; Li et al., 2026c). **Acc + Reg.** fits a predictor from the 128 semantic-mixture vectors to real target scores, searches for high predicted accuracy, and fills the selected cluster quotas uniformly. **Source + Reg.** instead optimizes shares over the 30 source branches, fitting a predictor of target-aligned merged-proxy loss and sampling uniformly within each source.

Within-cluster variant. CLIVO uses uniform sampling by default. CLIVO + ICONS preserves its learned quotas and selects the highest-scoring ICONS examples within each cluster. Global ICONS instead selects across the entire pool. Both use the same target-specific reference examples.

B.5 REPEATED RUNS

Random, Acc + Reg., Source + Reg., and CLIVO report three-run means; the other methods report one run. CLIVO’s repeats keep the selected mixture fixed and vary the final sampling/training seed. Table 7 gives the mean, sample standard deviation, and sample variance, using the $n - 1$ denominator. Table 8 reports the available baseline statistics: Random, Acc + Reg., and Source + Reg. use population standard deviation over three data-sampling seeds (42, 43, and 44).

Table 7: **CLIVO final-training repeats.** Three runs at fixed searched shares with uniform within-cluster sampling. Variance uses the $n - 1$ denominator.

Target	Mean	Standard deviation	Sample variance
General	67.28	0.53	0.28
Parsing	79.64	0.59	0.34
Knowledge	61.76	0.11	0.01
Overall	68.13	0.13	0.02

Table 8: **Baseline repeats.** Three-seed mean $\pm$ population standard deviation.

Method	Runs	Overall	General	Parsing	Knowledge
Random	3	66.61 $\pm$ 0.23	63.57 $\pm$ 0.28	77.73 $\pm$ 0.09	60.05 $\pm$ 0.44
Source + Reg.	3	67.00 $\pm$ 0.17	62.91 $\pm$ 0.39	79.12 $\pm$ 0.26	59.80 $\pm$ 0.16
Acc + Reg.	3	67.53 $\pm$ 0.05	65.44 $\pm$ 0.38	77.80 $\pm$ 0.03	61.40 $\pm$ 0.26

B.6 BENCHMARK-LEVEL RESULTS

Table 9 reports CLIVO’s three-run benchmark means and the single-run CLIVO + ICONS results. Each target has its own selected subset and trained model, so its benchmark profile is reported separately. Domain scores and the four-target Average are calculated before rounding.

Table 9: **Benchmark scores by target (%)**. CLIVO: three runs with random sampling; CLIVO + ICONS: one run, same quotas.

Target	Benchmark	CLIVO	CLIVO + ICONS
General	MMBench-EN	84.87	85.22
	MMStar	55.98	58.27
	GQA	60.99	61.32
Parsing	OCRBench	81.23	81.00
	ChartQA	78.04	78.92
Knowledge	MMLU	72.23	72.25
	MMLU-Pro	51.29	51.93
	MMBench-EN	84.03	84.43
Overall	MMStar	53.38	53.00
	GQA	59.87	60.68
	OCRBench	80.93	79.90
	ChartQA	77.01	77.28
	MMLU	70.74	70.71
	MMLU-Pro	50.93	50.38

B.7 ALLOCATION FREEDOM ON A FIXED EXPERT BANK

We keep the same 100 experts, initial mixture, validation objective, 120-update search, and final-training recipe. A nested Ward hierarchy of cluster centroids groups the experts into 20 or 50 parents; using 100 shares leaves every expert’s coefficient independently adjustable. For parent g , fix child ratios $a_{k|g} = w_k^0 / \sum_{h \in \mathcal{G}_g} w_h^0$. A zero-mass parent uses ratios proportional to child cluster sizes. The optimized parent shares $\mathbf{z}$ expand as $\mathbf{w} = \mathbf{B}\mathbf{z}$, where $B_{kg} = a_{k|g}$ within parent g and zero elsewhere.

Exact premerging identity. Define $\bar{\Delta}_g = \sum_{k \in \mathcal{G}_g} a_{k|g} \Delta_k$. Then

$$\tilde{\theta}(\mathbf{B}\mathbf{z}) = \theta_0 + \sum_g z_g \bar{\Delta}_g. \quad (11)$$

Thus tying fine coefficients is exactly equivalent to merging fixed coarse experts $\bar{\theta}_g = \theta_0 + \bar{\Delta}_g$. The parent gradients are $\mathbf{B}^\top \nabla_{\mathbf{w}} J_d$. This isolates allocation freedom with fixed experts.

Relation to briefly training a coarse expert. Let L_k be the mean training loss of child k and $L_g = \sum_k a_{k|g} L_k$. For one gradient-descent step from the common θ_0 , with common learning rate η ,

$$\sum_k a_{k|g} (\theta_0 - \eta \nabla L_k(\theta_0)) = \theta_0 - \eta \nabla L_g(\theta_0).$$

Thus, merging after one gradient-descent step exactly matches an update on the coarse mixture. For smooth losses and a fixed number T of small gradient-descent steps, $\theta_k^{(T)} = \theta_0 - T\eta \nabla L_k(\theta_0) + O(\eta^2)$. Averaging this expansion gives the coarse model’s first-order update, with a difference of order $O(\eta^2)$. The local approximation motivates premerging. Our experiment uses the exact identity in Eq. (11).

Table 10: **Allocation freedom.** J^* : minimum proxy loss; t^* : best step; C^* : 64-H100-equivalent hours; F : single-seed score. Bold: best F . Accounting: Section C.4.

Target	K	J^*	t^*	C^*	F
General	20	0.32639	120	0.662	67.197
	50	0.32801	80	≥ 0.343	66.833
	100	0.32145	100	0.808	67.017
Parsing	20	0.85952	0	0.00	79.810
	50	0.85965	100	0.833	80.120
	100	0.85866	120	0.993	80.260
Knowledge	20	0.68731	40	≥ 0.369	61.415
	50	0.68743	120	1.128	61.570
	100	0.68760	40	0.402	61.825
Overall	20	0.37324	40	0.360	67.994
	50	0.37097	100	0.865	67.846
	100	0.36904	100	0.818	68.021

Table 10 gives the full results underlying Figure 2. Each cell uses the same single-seed training protocol. Parsing and Knowledge improve with finer shares; General peaks at 20; Overall is not monotone.

B.8 EXPERT TRAINING BUDGET AND PROXY AGREEMENT

We train one expert per cluster from the common base on 64 H100 GPUs at a constant learning rate of 10^{-5} , with four packed sequences per GPU and a 16,384-token budget per sequence. We compare checkpoints indexed $b = 1, 5, 10$, holding this recipe fixed. The step-10 source checkpoints record 11 completed updates and a median of 16,591 processed examples per expert. A fully trained bank provides the reference. All banks use the same 128 mixtures and 10,000 examples.

Table 11: **Expert budget and two kinds of agreement.** Fidelity compares proxy with real-model loss over 128 mixtures. G/P/K/O are held-out correlations with real benchmark scores after budget-specific validation-cluster selection. Bold: highest in each correlation column.

Bank	Mean loss	Fidelity ρ	G ρ	P ρ	K ρ	O ρ
Step 1	1.283	0.485	0.383	0.435	0.523	0.367
Step 5	1.007	0.647	0.409	0.462	0.524	0.387
Step 10	0.972	0.730	0.518	0.448	0.527	0.460
Full schedule	0.940	0.644	0.577	0.382	0.440	0.457

Fidelity is the Spearman correlation between proxy and real-model mean losses over the 128 mixtures. For downstream prediction, five repeated four-fold splits use 96 fitting and 32 held-out probes. Within each fitting fold, clusters are ranked by the correlation between negative mean loss and target score; three-fold inner validation chooses the number retained from $\{1, 2, 4, 8, 16, 32\}$. The final columns report mean held-out correlations after budget-specific selection. The strongest proxy–real agreement and useful score prediction across targets support using step 10 as the shared expert bank.

B.9 GRADIENT SEARCH PROGRESS

The four searches reduce their frozen validation objectives (Figure 5 and Table 12). Full-panel evaluations every 20 updates select the best checkpoint, at steps 100, 120, 40, and 100 for General, Parsing, Knowledge, and Overall. General enters a near-best plateau at step 80, whereas losses for the other targets continue to fluctuate. Periodic evaluation keeps the best observed mixture throughout the 120-update search.

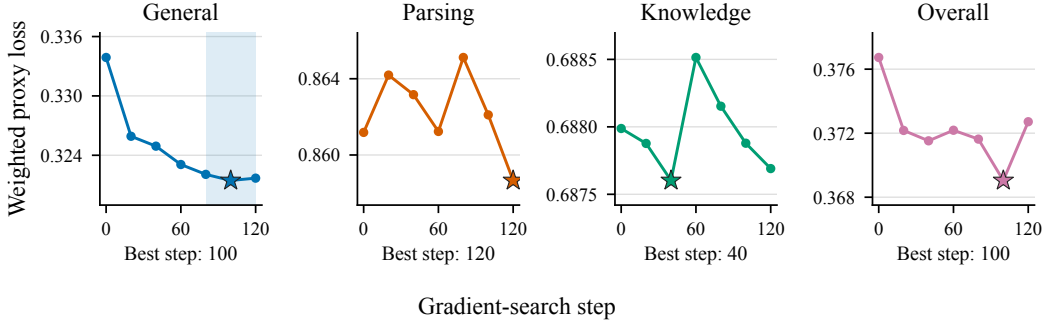


Figure 5: **Search progress on fixed validation objectives.** Stars mark the lowest recorded losses. General’s shaded region retains at least 95% of its observed loss decrease with two subsequent checks. Lines connect full-panel evaluations at 20-update intervals.

Table 12: **Search starts and checkpoints.** Full BF16 evaluation selects t^* ; remerged step-zero loss J_0 defines the decrease D_J .

Target	Start recipe	t^*	J_0	J^*
General	101	100	0.333885	0.321450
Parsing	119	120	0.861185	0.858662
Knowledge	27	40	0.687988	0.687602
Overall	101	100	0.376730	0.369045

The decrease in Table 3 is $D_J = 100(J_0 - J^*)/J_0$, using the same run’s evaluated start. Best-of-probe references are 66.600, 79.310, 61.465, and 67.9186 for General, Parsing, Knowledge, and Overall; the final score column uses three-run means. Table 5 lists the probe and final-model budgets.

C DATA INTERPRETATION, FLEXIBILITY, AND REUSE

C.1 DATA ANNOTATION AND COMPOSITION

We sample a 10,000-example panel across the 100 clusters for annotation with gpt-5.2-2025-12-11; the archive contains 9,555 valid outputs. The sample prompt labels task, content, format, and difficulty. The cluster prompt summarizes label frequencies and representative annotations. The response limits are 500 and 800 tokens, respectively; images use a maximum side length of 512 pixels and videos supply up to ten frames. An author-conducted check of sampled annotations yielded a human–GPT agreement rate of 0.92. These semantic labels support data visualization and interpretation. Table 13 gives both prompts.

Table 13: **GPT annotation prompts.** Sample labels describe task, content, format, and difficulty; cluster summaries aggregate these annotations.

Sample annotation. The interaction’s text and visual inputs follow this prompt.

You are an expert Data Annotation Specialist. Your task is to analyze User-Assistant interaction samples and classify them into 4 dimensions using STRICT standardized labels.

The data comes from diverse multi-modal sources including VQA, Document Reasoning, Mathematical Logic, Web Agents, and Grounding tasks. You must output a SINGLE valid JSON object.

1. Task Dimension (Hierarchical)
Classify the user’s primary intent.

- "task_category": Choose one of ["Perception", "Reasoning", "Generation", "Knowledge", "Chat"]
- "task_type": Choose the specific sub-type based on the category:
 - If Perception: ["OCR", "Object Detection", "Image Captioning", "Visual Grounding", "Video Tracking"]
 - If Reasoning: ["Math Solving", "Chart/Data Analysis", "Document Understanding", "Logic/Puzzle", "Web/UI Navigation", "Scientific Analysis"]
 - If Generation: ["Creative Writing", "Translation", "Summarization", "Code Implementation", "Long-form Content"]
 - If Knowledge: ["Fact Lookup", "Concept Explanation", "Trivia", "How-to Guide"]
 - If Chat: ["Roleplay", "General Chat", "Multi-turn Dialogue"]

2. Content Dimension (Visual & Semantic split)
Analyze the domain and visual inputs.

- "visual_content": Choose one: ["No Image", "Natural Scene", "Human/Face", "Document/Text", "Chart/Graph/Plot", "UI/Webpage", "Art/Illustration", "Abstract/Pattern"]
- "text_domain": Choose one: ["General/Daily Life", "STEM", "Humanities/Art", "Medical/Health", "Legal/Finance", "Code/Programming", "Web/Internet"]

3. Format Dimension (Input & Output Structure)
Analyze the structural form.

- "input_composition": Choose one: ["Text-Only", "Single-Image", "Multi-Image", "Video", "Interleaved-Text-Image"]
- "response_style": Choose one: ["Multiple Choice", "Yes/No", "Short Answer", "Long Explanation/CoT", "Code Block", "Structured Format (JSON/Table)", "Bounding Box/Coordinates"]

4. Difficulty Dimension (Cognitive Load)

- "difficulty_level": Choose one: ["Low", "Medium", "High"]

****CRITICAL:**** Output raw JSON only. No markdown. No explanations.

Continued on the next page.

Table 13: GPT annotation prompts (continued).

Cluster summary. Inputs follow under Cluster Aggregated Statistics (label histograms) and Sampled Annotations (up to 100 records).

You are an expert Data Analyst and Dataset Curator.
 I will provide you with the aggregated annotation statistics and a sample of individual interactions from a specific data cluster. These individual samples were previously annotated across 4 strict dimensions: Task, Content, Format, and Difficulty.

Your objective is to analyze this aggregated data and the samples to construct a comprehensive, multi-dimensional profile of this specific cluster.

Please output a valid JSON object strictly matching the following schema. Use the predefined label vocabularies from the sample annotations when identifying dominant categories.

```
{
  "task_dimension_summary": {
    "dominant_task_category": "The most frequent task category (e.g., Perception, Reasoning, etc.)",
    "top_task_types": ["List the top 1-3 specific task_types present in this cluster"],
    "task_diversity_note": "A 1-sentence analysis of whether this cluster is highly focused on one task or mixes several."
  },
  "content_dimension_summary": {
    "primary_visual_content": ["List the top 1-2 most frequent visual_content types"],
    "dominant_text_domain": "The most frequent text_domain",
    "content_fusion_analysis": "A 1-sentence description of how the visual and textual domains interact in this cluster."
  },
  "format_dimension_summary": {
    "dominant_input_composition": "The most frequent input_composition (e.g., Interleaved-Text-Image)",
    "typical_response_style": ["List the top 1-2 expected response_styles"]
  },
  "difficulty_dimension_summary": {
    "overall_difficulty_level": "Low / Medium / High / Mixed (based on majority distribution)",
    "cognitive_load_description": "A 1-sentence explanation of why the data requires this level of cognitive load."
  },
  "comprehensive_cluster_profile": {
    "short_definition": "A precise, 1-sentence title or definition for this cluster (e.g., 'Complex Scientific Document Reasoning and Formatting').",
    "detailed_description": "A comprehensive 3-4 sentence paragraph synthesizing all 4 dimensions. Describe the typical user intent, the nature of the multi-modal inputs, the expected output structure, and the overall technical or cognitive demands of the data within this cluster."
  }
}
```

****CRITICAL INSTRUCTIONS:****

1. Output raw JSON ONLY.
 2. Do NOT wrap the JSON in markdown code blocks (e.g., no ```json ... ```).
 3. Do NOT include any introductory or concluding text.
 4. Ensure all dimensional analyses strictly reflect the provided sample data and statistics.
-

From fine tasks to coarse capabilities. We map the first task in each cluster summary’s `top_task_types` to a coarse capability using Table 14. These annotations make the learned allocation interpretable; embeddings define clusters and the target objective sets their shares.

Table 14: **Coarse capability labels for visualization.** Each cluster’s primary annotated task determines its coarse label. The original task descriptions are retained for finer interpretation.

Coarse capability	Primary task types
Visual QA & Captioning	Image Captioning
Grounding & Detection	Object Detection; Visual Grounding; Video Tracking
OCR & Documents	OCR; Document Understanding
Charts & Data	Chart/Data Analysis
Math & Logic	Math Solving; Logic/Puzzle
Knowledge & Science	Scientific Analysis; Concept Explanation; Fact Lookup
Generation & Other	Remaining task types, including code and creative generation

Composition statistics and cases. For normalized shares, the effective cluster count is $K_{\text{eff}} = \exp(-\sum_{k:w_k>0} w_k \log w_k)$: 3.024, 10.134, 3.923, and 4.362 for General, Parsing, Knowledge, and Overall. Uniform allocation gives each cluster 1%; a category A therefore has uniform share $|A|/100$ and enrichment $(\sum_{k \in A} w_k)/(|A|/100)$. The within-capability cases pair c60/c38 for General, c27/c14 for Parsing, and c36/c20 for Knowledge. Percentages are shares of the total budget.

Table 15 compares the complete mixtures. Total variation is $\frac{1}{2} \sum_k |p_k - q_k|$; Jensen–Shannon divergence is $\frac{1}{2} \text{KL}_2(p||m) + \frac{1}{2} \text{KL}_2(q||m)$, where $m = (p + q)/2$.

Table 15: **Target-dependent mixtures.** TV and JSD compare final cluster shares.

Target pair	TV	JSD (bits)
General / Parsing	0.9783	0.9481
General / Knowledge	0.9999	0.9995
General / Overall	0.2610	0.2126
Parsing / Knowledge	0.9895	0.9789
Parsing / Overall	0.8823	0.8338
Knowledge / Overall	0.9765	0.9235

C.2 WHY ALLOCATION AND SAMPLE RANKING COMPLEMENT EACH OTHER

Any size- N subset $\mathcal{S}$ induces cluster counts $n_k = |\mathcal{S} \cap \mathcal{D}_k|$ and shares $r_k = n_k/N$. Its empirical sampling distribution factors as

$$p_{\mathcal{S}}(x) = r_k q_{\mathcal{S}}(x | k), \quad q_{\mathcal{S}}(x | k) = \frac{\mathbf{1}\{x \in \mathcal{S}\}}{n_k}, \quad x \in \mathcal{D}_k, n_k > 0. \quad (12)$$

Group-level allocation explicitly chooses the marginal shares r_k ; within-group selection determines which examples realize them. At the finest partition, every group contains one example, and a share of $1/N$ or zero specifies selection. Group allocation thus recovers sample-level selection.

For intermediate partitions, composition and ranking offer complementary control. Global top- N ranking maximizes the sum of sample scores and lets cluster shares follow from that ranking. Fixing learned quotas instead maximizes the same scores subject to $|\mathcal{S} \cap \mathcal{D}_k| = n_k$, solved by taking the top n_k examples in each cluster. Thus, quotas control target-relevant coverage while ranking chooses representatives within that coverage. CLIVO + ICONS implements this factorization, combining learned composition with an existing ranker (Table 1).

C.3 REUSING FEEDBACK ACROSS TARGETS AND GRANULARITIES

Let C_{shared} denote the construction cost of embeddings, clusters, experts, real probes, and the per-example proxy-loss cache. The incremental cost of selecting for a new target d is

$$\Delta C_{\text{target}}(d) = C_{\text{score}}(Y, d) + C_{\text{fit}}(Y, H, m_d) + C_{\text{grad}}(G, T, m_d). \quad (13)$$

The terms are benchmark evaluation of the Y saved probes, example screening and weight fitting on cached losses, and T search updates with G adjustable shares and m_d active validation examples. Expert/probe training and cache construction are reused. Final model training is a separate cost common to subset-selection methods.

For a new granularity at a fixed target, the incremental cost is $\Delta C_{\text{gran}}(G) = C_{\text{tie}}(G) + C_{\text{grad}}(G, T, m_d)$, where C_{tie} constructs the parent-to-child mapping. On the fixed $K_0 = 100$ expert bank, write $D = [\Delta_1, \dots, \Delta_{K_0}]$ and $\mathbf{w} = B_G \mathbf{z}$. A single accumulated backward pass supplies $\nabla_{\mathbf{z}} J_d = B_G^\top D^\top \nabla_\theta J_d$. For our implementation, the search cost decomposes as

$$C_{\text{grad}}(G, T, m_d) = T[C_{\text{merge}}(K_0) + C_{\text{FB}}(m_d) + C_{\text{directions}}(K_0) + C_{\text{update}}(G)] + C_{\text{check}}. \quad (14)$$

The dominant merge, forward/backward, and expert-direction computations are shared across the 20/50/100-share settings; only the coefficient mapping and update change. Finer allocation therefore adds no model backward pass per extra share and no higher-dimensional response fit. In comparison, a regression search that constructs Q new candidate labels incurs $Q C_{\text{label}} + C_{\text{regfit}}(Q, G)$ before proposal and recheck costs. Section C.4 reports the measured search workloads.

C.4 INCREMENTAL SEARCH BUDGET

We report compute as *64-H100-equivalent hours*: H100-equivalent GPU-hours divided by 64, using the same peak-BF16 normalization across runs. This is a common compute-budget unit. The measured workload includes iterative merging and model forward/backward computation; shared construction and final model training are excluded.

Regression reference. LightGBM predictors use nested budgets of 256, 512, 1,024, 2,048, and 4,096 mixtures labeled with proxy losses. Five-fold cross-validation Spearman reaches 0.935/0.909/0.898/0.885 at 4,096 labels for General/Parsing/Knowledge/Overall. The final doubling improves these values by 0.024/0.013/0.055/0.036, smaller than the preceding doubling for every target. We use these 4,096 shared labels as the high-budget reference.

Table 16: **Incremental search compute.** Gradient costs cover all 120 updates. Regression cost covers constructing the shared 4,096 proxy-loss labels. Units: 64-H100-equivalent hours.

Workload	Compute budget
Gradient: General	0.960
Gradient: Parsing	0.993
Gradient: Knowledge	1.123
Gradient: Overall	0.971
Gradient: four-target total	4.046
Regression: 4,096 shared labels	5.509

The four gradient searches use 26.6% less normalized compute than regression labeling. In Table 10, C^* measures cost to the first lowest-loss checkpoint. Upper endpoints summarize reconstructed timing bounds; $\geq$ denotes incomplete timing. Parsing with 20 shares retains its start, but the complete search costs 0.312 hours in the same normalized units.

Table 17: **Search endpoints.** Single-seed 100K scores. GD: gradient search; regression: 4,096 labeled mixtures.

Target	GD 20	GD 50	GD 100	Regression
General	67.20	66.83	67.02	67.09
Parsing	79.81	80.12	80.26	80.10
Knowledge	61.42	61.57	61.83	62.17
Overall	67.99	67.85	68.02	68.22

Regression endpoint protocol. Regression ranks 10 million proposals by predicted log loss. Its 4,096 labels cover 115 feasible probes, four gradient endpoints, 994 perturbations of gradient solutions, and 2,983 Dirichlet samples. Gradient endpoints contribute labels but are excluded from final selection. Table 17 reports single-seed scores for the selected candidates.