CLOSING THE CURATION LOOP: SEMANTIC COMPRESSION AND LEARNING-GUIDED EXPANSION FOR MLLMS MID-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Efficient mid-training of multimodal large language models (MLLMs) requires both reducing repetitive supervision and adapting training data to the model's changing learning needs. We propose a closed-loop data curation framework for image–text mid-training that combines coverage-preserving semantic compression with learning-guided expansion. First, we jointly assess image–text pair specificity and local semantic diversity in an MLLM-derived representation space, and use cluster-level budget allocation to construct an initial image–text training set with broad semantic coverage and reduced redundancy. We then evaluate the model's prediction loss across semantic regions using a fixed probe set, with reference-relative loss gaps serving as an empirical signal of learning potential to guide subsequent data acquisition. By alternating between model diagnosis, data acquisition, and continued mid-training, the framework adapts the training data mixture to the model's learning state while preserving semantic coverage. Under the same sample budget, our framework achieves better overall performance than existing data-selection methods across multiple widely used multimodal image–text understanding benchmarks. Ablation studies further support the complementary roles of semantic compression and learning-guided expansion.

1 INTRODUCTION

Mid-training multimodal large language models (MLLMs) (Liu et al., 2023a; Qwen Team, 2025) aims to align vision and language and learn to describe visual entities, attributes, and relations. Effective data selection depends on both the complementary supervision offered by different samples and its relevance to the model's changing learning needs. This motivates two connected questions: *How can we build an initial training set with broad semantic coverage and little redundancy? How can we add useful supervision as the model learns?* Together, these questions call for a data-curation loop guided by model learning.

Existing multimodal data curation pipelines primarily assess image–text consistency and improve the training distribution through concept rebalancing and semantic deduplication (Mahmoud et al., 2024; Xu et al., 2025; An et al., 2025; Abbas et al., 2023). However, high image–text matching scores and broad visual-concept coverage do not necessarily ensure diverse supervision within a limited training budget. Many samples may still describe similar attributes, relations, and scenes. Our preliminary experiments reveal clear density differences in the joint image–text semantic space even after visual-concept rebalancing: larger semantic clusters tend to have higher nearest-neighbor similarity, suggesting that local semantic redundancy may persist. Building an efficient initial training set therefore requires considering both which samples provide less repetitive supervision and how many samples to retain from each semantic region, reducing repetition in frequent patterns while preserving overall semantic coverage.

As training progresses, data value also depends on the model's current learning state. Studies of data selection based on reducible loss and learning progress show that model feedback can inform the prioritization of training samples (Mindermann et al., 2022; Evans et al., 2024; Yang et al., 2026). Since the model may learn supervision from different semantic regions to different degrees, an initial subset with broad coverage and little redundancy does not necessarily retain an optimal semantic

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.

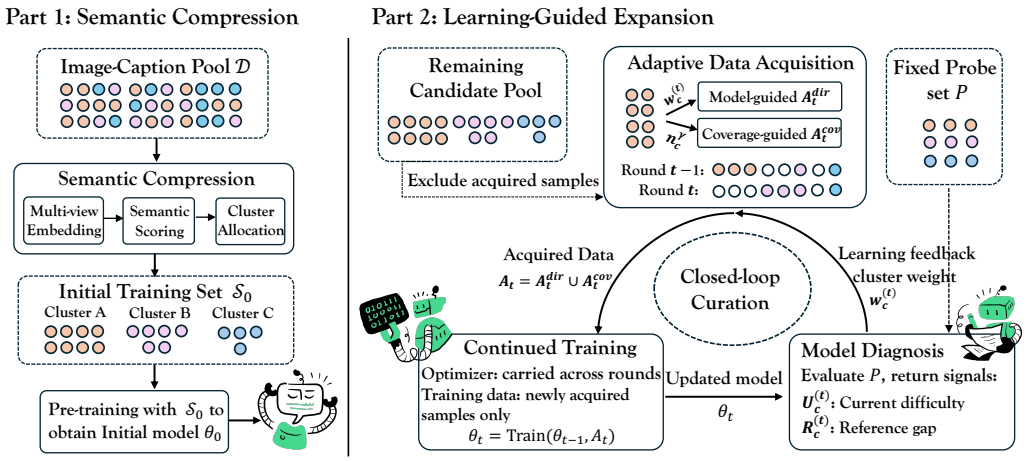


Figure 1: Overview of the closed-loop data curation framework. Semantic compression constructs the initial training set while preserving semantic coverage. Subsequent rounds alternate between model diagnosis, acquisition of new data from the candidate pool, and continued mid-training, with the updated model guiding the next round of selection.

mixture throughout training. Initial compression should therefore be viewed as a starting point rather than a final judgment of data value. This motivates allocating subsequent training budgets to semantic regions where learning gaps remain and further training may be beneficial. By acquiring additional data from the candidate pool over successive rounds, we explore adaptive adjustment of the semantic mixture for subsequent training, closing the loop between data selection and model learning.

We propose a closed-loop data curation framework for MLLM image–text mid-training that combines coverage-preserving semantic compression with learning-guided data expansion (Figure 1). First, an MLLM-based embedding model produces image, text, and joint image–text representations. We measure pair specificity by comparing the original match with semantically similar alternative pairings, and local semantic diversity through neighbor distances in the joint representation space. Cluster-level budget allocation and within-cluster ranking then construct the initial training set. After initial mid-training, we diagnose the model using a fixed probe set: prediction loss identifies difficult semantic regions, while the loss gap relative to a reference model serves as an empirical signal of further learning potential. These signals determine cluster-level sampling proportions, which are combined with cluster-size-based sampling to preserve semantic coverage. The model continues mid-training on newly acquired data, and its updated learning state guides the next round of selection, forming a diagnose–acquire–train loop within a fixed total data budget.

Our ablations show that static compression does not consistently improve robustness, while combining diagnosis-guided acquisition with semantic coverage yields stronger results across all four evaluation groups than either strategy alone. Our contributions are threefold:

- We propose coverage-preserving semantic compression that jointly assesses pair specificity and local semantic diversity, and uses cluster-level budget allocation to reduce repetitive supervision while preserving semantic coverage. This balances sample-level supervision complementarity with coverage across semantic regions.
- We establish a learning-feedback-driven data selection loop that combines current prediction difficulty with reference-relative loss gaps to adjust the semantic composition of acquired data across rounds, adapting the data mixture to the model's learning state within a fixed sample budget.
- We validate the framework's overall performance advantage and key mechanisms. With 5M samples, it achieves the highest average score of 64.7 across eight benchmarks among the evaluated methods. Ablations support the complementary roles of semantic coverage and model feedback, as well as the two diagnostic signals.

2 CLOSED-LOOP DATA CURATION

Given a candidate image–caption dataset $\mathcal{D} = \{(I_i, T_i)\}_{i=1}^{N}$ and a data budget B, we formulate data curation as a closed-loop process. We first construct an initial training set $\mathcal{S}_0$ according to the intrinsic semantic structure of the data, and then iteratively expand it from the candidate pool $\mathcal{R}_0 = \mathcal{D} \setminus \mathcal{S}_0$ based on the model's learning feedback.

2.1 EMPIRICAL MOTIVATION

Existing CLIP-based multimodal data curation methods primarily focus on global image–text consistency and visual-concept coverage (An et al., 2025). However, MLLM mid-training also involves learning fine-grained supervision from image–caption pairs, including entity attributes, object relations, event semantics, and description patterns. To characterize such training signals, we use multimodal embedding models derived from MLLMs (e.g., Jina-embeddings-v5-omni (Hönicke et al., 2026) and Qwen3-VL-Embedding (Li et al., 2026)) to embed the data and analyze its joint image–text semantic structure. We sample 10M examples from LLaVA-OneVision-1.5-Mid-Training-85M in proportion to the original data sources.

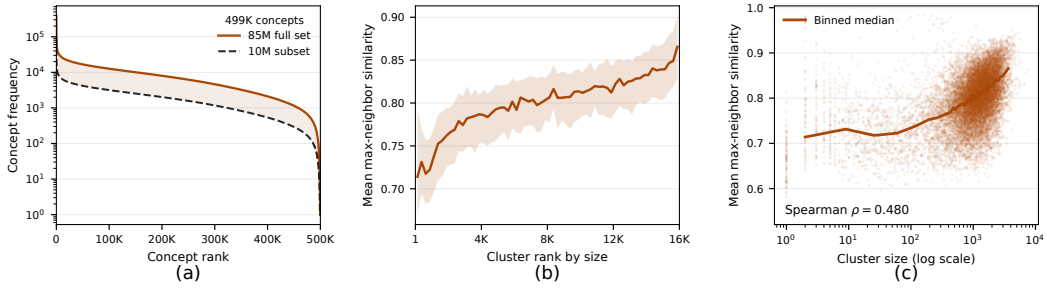


Figure 2: Visual-concept coverage and joint image–text semantic structure. **(a)** Ranked concept frequencies in the 10M subset and 85M pool. **(b)** Mean maximum-neighbor similarity, a proxy for semantic density, across 64 equal-count cluster-size bins; the curve and shading show medians and interquartile ranges. **(c)** Cluster size versus this density proxy for all 16,384 clusters, with a binned-median trend and Spearman correlation.

Figure 2(a) shows that the subset preserves the relatively balanced visual-concept distribution of the full data pool. Figure 2(b) shows that larger semantic clusters tend to be internally denser: from the smallest to the largest cluster-size decile, the median maximum-neighbor similarity increases by 15.6%. The cluster-level scatter in Figure 2(c) further supports this positive association (Spearman $\rho = 0.480$). Large clusters therefore aggregate more image–caption pairs that express similar entity attributes, spatial relations, and description patterns. These results show that visual-concept balance does not imply balance in the joint image–text semantic space. Data curation must therefore compress repetitive supervision in frequent semantic regions while increasing the relative coverage of long-tail semantic regions.

2.2 COVERAGE-PRESERVING SEMANTIC COMPRESSION

Based on the above observation, we first perform semantic compression in an MLLM-based embedding space to reduce repetitive supervision in frequent semantic regions while preserving overall semantic coverage. The overall semantic compression framework is illustrated in Figure 3.

Multi-view semantic neighborhoods. After excluding invalid samples that cannot be parsed or have missing fields through quality filtering (see Appendix A.1), we use the same MLLM-based multimodal embedding model E to separately encode the image, the caption, and the complete image–caption pair, producing three sets of vectors:

$$m_i = \mathrm{Norm}(E(I_i, T_i)), \qquad v_i = \mathrm{Norm}(E(I_i)), \qquad t_i = \mathrm{Norm}(E(T_i)), \tag{1}$$

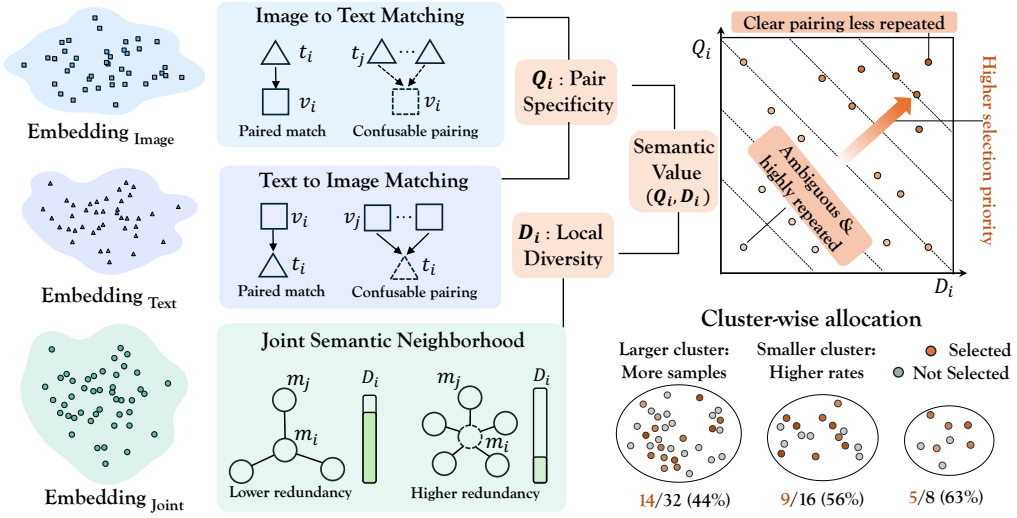


Figure 3: Semantic compression overview. Bidirectional image–text matching and joint semantic neighborhoods yield the percentile-normalized pair-specificity score Q_i and local-diversity score D_i, respectively. Their combination guides within-cluster ranking, while cluster-level budgets preserve semantic coverage.

where $\mathrm{Norm}(\cdot)$ denotes L_2 normalization. The joint representation m_i is used to partition the data into semantic clusters $\{C_c\}$ and retrieve global neighbors $\mathcal{N}_K(i)$ without restricting the search to cluster boundaries, while the image representation v_i and text representation t_i are used to evaluate cross-modal correspondence.

Semantic supervision value. We first assess whether the original image–text pair has a clear matching advantage over semantically similar alternatives. Its similarity $s_{ii} = v_i^\top t_i$ alone does not reveal whether the image also matches other captions or the caption also fits other images. Following hard-negative visual–semantic matching (Faghri et al., 2018), we perform bidirectional matching within the assigned semantic cluster and select the most similar alternative captions and images as hard negatives. Their mean similarities to the anchor image and caption are denoted by h_i^T and h_i^I, respectively. We define pair specificity as

$$q_i = s_{ii} - \frac{1}{2}\left(h_i^T + h_i^I\right). \tag{2}$$

A larger q_i indicates that the original pair is less easily confused with these alternatives, whereas a smaller value indicates a weaker matching advantage. This score measures relative pair specificity rather than requiring a strict one-to-one correspondence. However, even a distinctive pair may provide supervision already repeated by other complete image–text pairs. We therefore assess local replaceability in the joint semantic neighborhood using the mean cosine distance to the nearest neighbors:

$$d_i = \frac{1}{K}\sum_{j\in\mathcal{N}_K(i)}\left(1 - m_i^\top m_j\right). \tag{3}$$

A larger d_i indicates greater joint-semantic differences between the sample and its neighbors, but does not by itself establish the value of that sample. We transform q_i into a global percentile Q_i and d_i into a within-cluster percentile D_i, and define its model-independent semantic score as

$$S_i^{\mathrm{sem}} = \lambda Q_i + (1-\lambda) D_i, \tag{4}$$

where $\lambda = 0.5$ by default. This score combines the matching advantage of the original pair with its semantic distinctiveness from neighboring image–caption pairs; final selection also depends on the cluster-level budgets below.

Cluster-level allocation and candidate pool. To prevent global Top-B_0 selection from being dominated by large clusters under a limited budget, we allocate the initial budget according to cluster

Algorithm 1 Learning-guided expansion

Input: $\theta_0, \mathcal{E}, \mathcal{P}, \ell^{\text{ref}}, S^{\text{sem}}, T, \{\Delta B_t\}, \rho$
for $t = 1, \ldots, T$ **do**
  $(U^{(t)}, R^{(t)}) \leftarrow \textsc{Diagnose}(\theta_{t-1}, \mathcal{P}, \ell^{\text{ref}})$
  $\mathcal{G} \leftarrow \text{Top}_\rho(U^{(t)}) \cap \{c : |\mathcal{E}_c| > 0\}$
  $w \leftarrow 1 + \text{Percentile}_{\mathcal{G}}(R^{(t)})$
  $b \leftarrow \textsc{Allocate}(\mathcal{E}, \mathcal{G}, w, \Delta B_t/2)$
  $\mathcal{A}^{\text{dir}} \leftarrow \bigcup_{c\in\mathcal{G}} \text{Top}_{b_c}(\mathcal{E}_c; S^{\text{sem}})$
  $\mathcal{E} \leftarrow \mathcal{E} \setminus \text{Excl}(\mathcal{A}^{\text{dir}})$
  $\mathcal{A}^{\text{cov}} \leftarrow \textsc{Compress}(\mathcal{E}, \Delta B_t - |\mathcal{A}^{\text{dir}}|)$
  $\mathcal{E} \leftarrow \mathcal{E} \setminus \text{Excl}(\mathcal{A}^{\text{cov}})$
  $\mathcal{A}_t \leftarrow \mathcal{A}^{\text{dir}} \cup \mathcal{A}^{\text{cov}}$
  $\theta_t \leftarrow \textsc{Train}(\theta_{t-1}, \mathcal{A}_t)$
end for
return θ_T

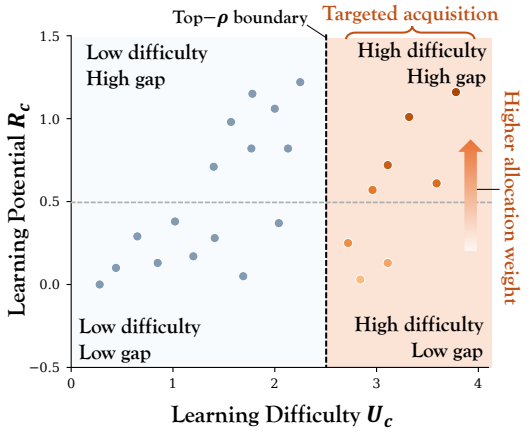


Figure 4: Learning-guided expansion. Left: closed-loop selection and training. Right: cluster diagnosis (schematic), where U_c determines targeted eligibility and R_c adjusts sampling weights. Each point is a cluster; the horizontal split is illustrative.

size:

$$\widetilde{b}_c^0 = B_0 \frac{n_c^\gamma}{\sum_g n_g^\gamma}, \tag{5}$$

where $\gamma \in [0, 1]$ controls the trade-off between data scale and semantic coverage, and is set to $\gamma = 0.7$ by default. After converting it into integer budgets b_c^0 that satisfy the cluster-capacity and total-budget constraints, we obtain

$$\mathcal{S}_0 = \bigcup_c \text{Top}_{b_c^0}\{i \in C_c \mid S_i^{\text{sem}}\}, \qquad \mathcal{R}_0 = \mathcal{D} \setminus \mathcal{S}_0. \tag{6}$$

After ranking samples within each cluster according to their semantic supervision value, cluster-level allocation controls the number of samples retained in each semantic region. The samples not selected in the first stage form the candidate pool $\mathcal{R}_0 = \mathcal{D} \setminus \mathcal{S}_0$. Importantly, this does not imply that these samples are valueless: static semantic structure only reflects similarities among samples and cannot determine whether they contain supervision that the current model has not yet learned. The subsequent closed loop therefore uses model learning states to iteratively acquire new training data from $\mathcal{R}_0$.

2.3 Learning-Guided Closed-Loop Expansion

Semantic compression determines the initial training distribution from the data structure, while subsequent data allocation adapts to the model's learning state. After initial mid-training on $\mathcal{S}_0$ to obtain θ_0, our core approach is to cycle through diagnosis, data sampling, and continued mid-training over multiple rounds. The detailed algorithm is presented in Figure 4 (left).

Probe set construction. To estimate learning states across semantic regions, we construct a fixed probe set $\mathcal{P}$ from the full candidate pool. Specifically, we allocate a 500K-sample budget evenly across semantic clusters and use pair-specificity scores to guide selection within each cluster, aiming to cover both inter- and intra-cluster diversity. Probes and their near-duplicates are excluded from target-model training to avoid biasing the diagnosis through direct training on these samples.

Cluster-level diagnosis. Diagnosis should capture both the model's current predictive performance and the potential value of further learning. For the former, at round t we use the current model θ_{t-1} to compute the probe language-modeling loss $\ell_i^{(t)} = \ell(\theta_{t-1}; x_i)$. However, a high loss may also arise from noisy or difficult samples and cannot by itself determine which learning gaps to address or the gains from further training. We therefore introduce a reference model (Mindermann et al., 2022) to assess learning potential. The loss gap between the current and reference models on the same data provides an empirical basis for this assessment.

Specifically, we divide the probe set into two cluster-balanced folds and perform lightweight LoRA adaptation on each fold separately. Each probe's reference loss ℓ_i^{ref} is computed by the model adapted on the opposite fold. This cross-fitting ensures that the reference model has not been trained directly on the scored sample, reducing bias from direct fitting. We then aggregate both signals by semantic cluster. Let $\mathcal{P}_c$ denote the probes in cluster c. We define current prediction difficulty and the relative reference gap as:

$$U_c^{(t)} = \frac{1}{|\mathcal{P}_c|} \sum_{i \in \mathcal{P}_c} \ell_i^{(t)}, \qquad R_c^{(t)} = \frac{1}{|\mathcal{P}_c|} \sum_{i \in \mathcal{P}_c} \big(\ell_i^{(t)} - \ell_i^{\text{ref}}\big). \tag{7}$$

The taxonomy in Figure 4 (right) illustrates their complementary roles: $U_c^{(t)}$ identifies semantic regions that remain difficult for the current model to predict, while $R_c^{(t)}$ serves as an empirical signal of their learning potential.

Diagnosis-guided data sampling. We use model diagnosis to adjust the proportions of semantic clusters in the supplementary data. We first rank clusters with valid current losses by $U_c^{(t)}$, retain the top fraction ρ, and remove those with no available candidates to obtain $\mathcal{G}_t$. Within this set, we convert $R_c^{(t)}$ into percentiles $p_c^{(t)}$ and define allocation weights:

$$w_c^{(t)} = 1 + p_c^{(t)}, \qquad c \in \mathcal{G}_t. \tag{8}$$

We reuse the normalized budget allocation in Eq. (5), replacing the size weights n_c^γ with $w_c^{(t)}$ and allocating over $\mathcal{G}_t$. Thus, $U_c^{(t)}$ determines the regions to prioritize, while $R_c^{(t)}$ adjusts their relative sampling proportions. After integer rounding and capacity constraints, we rank samples within each cluster by S_i^{sem} to obtain $\mathcal{A}_t^{\text{dir}}$.

We assign half of the round budget to diagnosis-guided selection, with $B_{\text{dir}}^{(t)} = \Delta B_t/2$. The remaining budget follows the cluster-size allocation and within-cluster ranking rules of semantic compression, applied to the remaining candidates, to obtain $\mathcal{A}_t^{\text{cov}}$ and maintain overall semantic coverage.

Continued mid-training and feedback. We merge the two sets and train only on the newly acquired data:

$$\mathcal{A}_t = \mathcal{A}_t^{\text{dir}} \cup \mathcal{A}_t^{\text{cov}}, \qquad \theta_t = \text{Train}(\theta_{t-1}, \mathcal{A}_t). \tag{9}$$

Each round inherits both the model weights and optimizer state from the previous stage and does not replay earlier data. The updated model re-evaluates the same probe set to guide the next round's budget allocation. By default, we add the same number of samples per round over multiple rounds, followed by a common instruction-tuning stage. The budget is measured by sample count; training settings are provided in Appendix C.

3 Experiments

3.1 Experimental Setup

Datasets and Model. The base data pool is a 10M-sample subset drawn from LLaVA-OneVision-1.5 An et al. (2025). The SFT data are sourced from FineVision Wiedmann et al. (2025), with category-balanced sampling across nine task categories yielding 3M samples. Our MLLM comprises a Qwen3-4B language model, a SigLIP2 vision encoder, and a randomly initialized visual merger Qwen Team (2025). During alignment pre-training, we optimize only the merger on LLaVA-558K Liu et al. (2023a); the resulting checkpoint serves as the common initialization for all subsequent experiments. During mid-training (MT) and supervised fine-tuning (SFT), all model parameters are unfrozen for training. Full hyperparameters and training configurations are provided in Appendix C.

Baselines and Evaluation. We compare against random sampling; heuristic selection based on CLIP-Score, perplexity Marion et al. (2023), and sequence length Zhao et al. (2024); and embedding-based data selection methods, including COINCIDE Lee et al. (2024), D^2-Pruning Maharana et al. (2024), DataTailor Yu et al. (2025), and PRISM Bi et al. (2026). We evaluate general multimodal understanding on MMBench-V1.1 Liu et al. (2023b), MMStar Chen et al. (2024), AI2D-TEST Kembhavi et al. (2016), MME Fu et al. (2023), and OCRBench Liu et al. (2023c); text and

Table 1: Comparison using 5M mid-training samples. Each *Avg.* row reports the arithmetic mean within its category. *Overall* averages the eight benchmarks and excludes MME. MME Total sums the perception (P) and reasoning (R) scores. Scores use one decimal place; bold and underline mark the best and second-best distinct results before rounding.

		Heuristic-based			*Embedding-based*				**Ours**	
Eval set	Random	CLIP-Score	Perplexity *arXiv'23*	Length *ICML'24*	COINCIDE *EMNLP'24*	D^2-Pruning *ICLR'24*	DataTailor *ICCV'25*	PRISM *EMNLP'26*	Compression	Closed loop
General										
MMBench	67.3	67.0	68.4	68.4	67.1	68.8	**68.8**	68.5	68.5	68.3
MMStar	44.3	43.7	44.1	45.2	43.5	44.4	44.7	44.4	45.3	**45.7**
OCRBench	56.0	56.9	**57.4**	56.8	55.2	57.2	53.6	55.2	57.2	**57.4**
AI2D_TEST	70.0	69.5	69.7	70.3	70.0	70.1	70.7	69.4	70.7	**70.8**
Avg.	59.4	59.3	59.9	60.2	59.0	60.1	59.4	59.4	60.4	**60.6**
Robustness										
HallusionBench	57.1	57.3	56.4	57.7	55.4	56.6	56.4	55.0	57.3	**60.1**
POPE	86.3	86.5	87.5	87.4	87.7	87.7	88.1	**88.3**	86.3	87.4
Avg.	71.7	71.9	71.9	72.6	71.6	72.1	72.2	71.7	71.8	**73.8**
Text-rich										
TextVQA	62.0	58.3	61.7	62.6	61.8	60.7	51.7	57.1	63.2	**64.1**
ChartQA	62.9	62.7	63.5	63.3	60.7	63.5	**64.7**	61.4	63.2	64.0
Avg.	62.5	60.5	62.6	62.9	61.3	62.1	58.2	59.2	63.2	**64.1**
MME										
MME-P	1500.2	1471.3	1507.4	1529.0	1507.0	1510.7	1479.0	1475.4	**1545.5**	1480.4
MME-R	324.3	289.6	321.4	336.4	324.6	335.0	321.8	330.7	340.4	**341.1**
MME Total	1824.5	1761.0	1828.8	1865.4	1831.7	1845.7	1800.8	1806.1	**1885.8**	1821.5
Overall	63.2	62.7	63.6	64.0	62.7	63.6	62.3	62.4	64.0	**64.7**

document understanding on ChartQA Masry et al. (2022) and TextVQA Singh et al. (2019); and object hallucination and visual hallucination on POPE Li et al. (2023) and HallusionBench Guan et al. (2024), respectively.

3.2 Main Results

Table 1 compares our method with a range of data-selection baselines, including random sampling, heuristics based on image–text similarity, perplexity, and sequence length, and recent representation-based selection methods. Under the same mid-training budget of 5M samples, our closed-loop data curation method achieves an average score of 64.7 across eight non-MME benchmarks, exceeding random sampling at 63.2 and the strongest baseline at 64.0. Compared with selecting the entire training subset in a single pass, these results suggest that dynamically adjusting the semantic composition of subsequent training data according to the model's current learning state can further improve overall performance within a fixed data budget.

These gains extend across multiple multimodal capabilities. Specifically, the closed-loop method achieves 45.7 on MMStar, 70.8 on AI2D-TEST, 60.1 on HallusionBench, 64.1 on TextVQA, and 57.4 on OCRBench. Comparing semantic compression with closed-loop expansion further shows that learning feedback provides additional gains beyond a strong initial subset: HallusionBench increases from 57.3 to 60.1, and TextVQA from 63.2 to 64.1. Although the closed-loop method does not achieve the highest score on every individual benchmark, its higher overall average indicates that learning-guided expansion benefits multiple capability dimensions rather than a single task category.

The gains from closed-loop expansion are not uniform across all metrics. Although it achieves the highest MME-R score, its MME-P and total scores remain below those of semantic compression alone. Our semantic compression method, meanwhile, achieves the best MME-P and MME

Experiments	*Acquisition*	
	Coverage	Feedback
A: Random acquisition	–	–
B: Coverage-preserving	✓	–
C: Diagnosis-guided	–	✓

Experiments	Coverage	*Feedback signals*	
		U_c	R_c
D: Prediction loss	✓	✓	–
E: Reference gap	✓	–	✓
F: Loss + ref. gap	✓	✓	✓

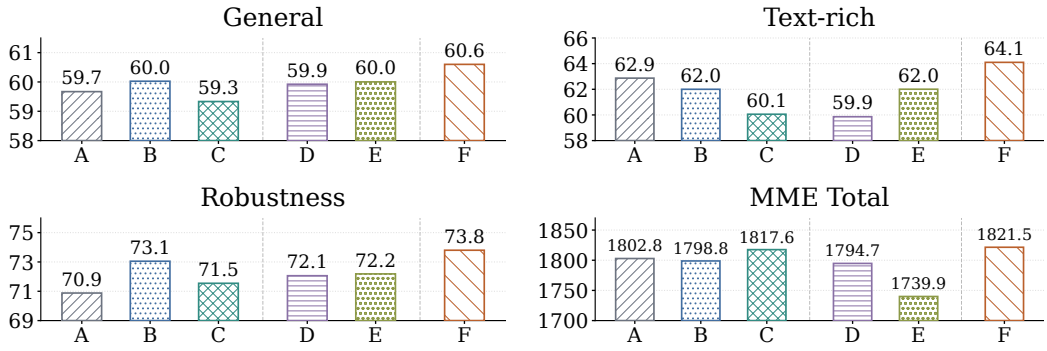


Figure 5: Ablations of data acquisition strategies and cluster-level learning diagnosis. Coverage denotes acquisition using the cluster-size-based budget allocation and within-cluster semantic ranking rules from Section 2.2. The full method (F) combines both acquisition strategies and both diagnostic signals.

Total scores among all evaluated methods. Overall, the advantage of closed-loop expansion lies primarily in its aggregate performance across the eight non-MME benchmarks, rather than consistent improvements on every metric.

3.3 Effectiveness of Closed-Loop Expansion

Model feedback benefits from semantic coverage. As shown in Figure 5, coverage-preserving acquisition alone (B) has limitations: it clearly improves Robustness but underperforms random acquisition on Text-rich and MME. Conversely, diagnosis-guided acquisition alone (C) improves MME but falls below random acquisition on General and Text-rich. Acquiring data from regions identified as valuable by model diagnosis is therefore insufficient to ensure balanced gains across capabilities. Combining the two strategies, the full method (F) achieves the highest scores across all four evaluation groups. This comparison supports treating targeted acquisition and coverage preservation as complementary mechanisms: the former responds to the model's current learning needs, while the latter maintains broader training opportunities, helping prevent subsequent data allocation from becoming overly concentrated on a small set of prioritized regions.

Prediction difficulty and learning potential are not interchangeable. Figure 5 shows that using prediction loss alone (D) yields a Text-rich score of only 59.9. Using the reference gap alone (E) raises this score to 62.0 but leads to a substantial drop in MME. The reference gap is therefore not a simple substitute for prediction loss and is better used alongside current prediction difficulty: the former provides an indication of relative room for improvement, while the latter identifies regions that remain difficult for the current model. Their joint use (F) improves all four evaluation groups over either signal alone, suggesting that effective acquisition requires identifying not only where difficulties remain, but also which difficulties warrant priority in further training.

3.4 Ablations and Analysis

Pair specificity and supervision diversity should be considered jointly. Table 2 shows that neither pair specificity alone nor local diversity alone yields the strongest performance across the evaluation suite. Their combination achieves the best results on General, Text-rich, and MME. This sup-

Table 2: Ablation of semantic supervision scoring for 5M-sample compression. All settings share the cluster budgets ($b_k \propto n_k^{0.7}$), candidate pool, filtering strategy, and training protocol. Checkmarks indicate pair specificity (Q) and local semantic diversity (D); any shared quality gate remains fixed.

Experiments	Semantic scoring		Benchmark results ↑			
	Q	D	General	Text-rich	Robustness	MME Total
Within-cluster random	–	–	59.39	<u>62.46</u>	<u>72.27</u>	1824.47
Pair specificity only	✓	–	<u>59.68</u>	60.01	71.45	1805.45
Local diversity only	–	✓	59.45	61.71	**72.30**	<u>1835.64</u>
Combined semantic scoring	✓	✓	**60.45**	**63.18**	71.81	**1885.81**

Table 3: Ablation of cluster-level budget allocation. All settings select 5M samples using the same candidate pool, semantic scores, filtering strategy, and training protocol. Global selection ranks all candidates without cluster quotas; the other settings allocate cluster budgets before within-cluster ranking.

Experiments	Budget allocation	Benchmark results ↑			
		General	Text-rich	Robustness	MME Total
Global selection	No cluster quotas	59.83	<u>61.76</u>	**71.95**	1838.36
Size-proportional	$b_c \propto n_c$	<u>60.00</u>	60.98	71.40	<u>1875.71</u>
Coverage-preserving	$b_c \propto n_c^{0.7}$	**60.45**	**63.18**	<u>71.81</u>	**1885.81**

ports jointly considering whether an image and its caption have a clear correspondence and whether similar supervision is already sufficiently represented in the data.

Selecting high-scoring samples requires quotas across semantic regions. As shown in Table 3, allocating budgets in proportion to cluster size improves General and MME over global selection, but reduces Text-rich from 61.76 to 60.98. In contrast, sublinear allocation outperforms size-proportional allocation across all four evaluation groups, raising Text-rich to 63.18. Cluster quotas thus do more than constrain the selection scope: moderately increasing the relative retention rate of smaller clusters is more beneficial in this setting than directly preserving the original cluster-size proportions.

Semantic compression cannot fully replace model feedback. Combined semantic scoring achieves a Robustness score of 71.81, below the 72.27 of random selection; coverage-preserving budget allocation also falls short of the 71.95 achieved by global selection. These results show that the benefits of static semantic selection do not extend to every capability, so compression should not be treated as a final judgment of data value. This motivates subsequent model-guided acquisition: candidate regions should be reassessed according to the model's learning state to address learning needs that semantic structure alone may not reveal.

4 Conclusion

We propose a closed-loop data curation framework for MLLM image–text mid-training that combines coverage-preserving semantic compression with learning-guided expansion. Semantic compression constructs an initial training set with reduced redundancy and broad coverage, while subsequent expansion uses model diagnosis to adapt the semantic mixture across training rounds. Experiments show improved overall performance under the same sample budget, and ablations support the complementary roles of semantic coverage and model feedback. These findings suggest that efficient data curation depends not only on which samples are retained before training, but also on how subsequent data selection adapts to the model's learning state. Future work will investigate the computational overhead of diagnosis and reference models, and evaluate the framework with larger models and more diverse multimodal data.

AI USE STATEMENT

Generative AI tools (Codex/DeepSeek/GLM) were used to polish author-written text for clarity and language and to generate portions of the research code. They were not used to formulate the research hypotheses, design the methodology or experiments, derive mathematical claims, or interpret the experimental results. All AI-assisted text was reviewed and revised by the authors. All AI-generated code was manually reviewed and validated against the intended algorithms before being used in the experiments. The authors made all final scientific and engineering decisions and take full responsibility for the content of this work.

REFERENCES

Amro Kamal Mohamed Abbas, Kushal Tirumala, Daniel Simig, Surya Ganguli, and Ari S. Morcos. SemDeDup: Data-efficient learning at web-scale through semantic deduplication. In *ICLR 2023 Workshops: MRL*, 2023. URL https://mlanthology.org/iclrw/2023/abbas2023iclrw-semdedup/.

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, Chunsheng Wu, Huajie Tan, Chunyuan Li, Jing Yang, Jie Yu, Xiyao Wang, Bin Qin, Yumeng Wang, Zizhen Yan, Ziyong Feng, Ziwei Liu, Bo Li, and Jiankang Deng. LLaVA-OneVision-1.5: Fully open framework for democratized multimodal training, 2025. URL https://arxiv.org/abs/2509.23661.

Jinhe Bi, Aniri, Zengjie Jin, Yifan Wang, Danqi Yan, Wenke Huang, Xiaowen Ma, Sikuan Yan, Artur Hecker, Mang Ye, Xun Xiao, Hinrich Schuetze, Volker Tresp, and Yunpu Ma. PRISM: Self-pruning intrinsic selection method for training-free multimodal data selection, 2026. URL https://arxiv.org/abs/2502.12119.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024. URL https://arxiv.org/abs/2403.20330.

Talfan Evans, Nikhil Parthasarathy, Hamza Merzić, and Olivier J. Hénaff. Data curation via joint example selection further accelerates multimodal learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4485.

Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. VSE++: Improving visual-semantic embeddings with hard negatives. In *British Machine Vision Conference*, pp. 12. BMVA Press, 2018. URL https://bmva-archive.org.uk/bmvc/2018/contents/papers/0344.pdf.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. MME: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. URL https://arxiv.org/abs/2306.13394.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. HallusionBench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Florian Hönicke, Michael Günther, Andreas Koukounas, Mohammad Kalim Akram, Scott Martens, Saba Sturua, and Han Xiao. jina-embeddings-v5-omni: Geometry-preserving embeddings via locked aligned towers. *arXiv preprint arXiv:2605.08384*, 2026. URL https://arxiv.org/abs/2605.08384.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European Conference on Computer Vision*, 2016. URL https://arxiv.org/abs/1603.07396.

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. Association for Computational Linguistics, November 2024. doi: 10.18653/v1/2024.emnlp-main.291.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Qwen3-VL-Embedding and Qwen3-VL-Reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026. URL https://arxiv.org/abs/2601.04720.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023a. URL https://arxiv.org/abs/2304.08485.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. MMBench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023b. URL https://arxiv.org/abs/2307.06281.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xucheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. OCRBench: On the hidden mystery of OCR in large multimodal models. In *arXiv preprint arXiv:2305.07895*, 2023c. URL https://arxiv.org/abs/2305.07895.

Adyasha Maharana, Prateek Yadav, and Mohit Bansal. D^2 pruning: Message passing for balancing diversity and difficulty in data pruning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/f646a4c970799f789e031b63161d018d-Abstract-Conference.html.

Anas Mahmoud, Mostafa Elhoushi, Amro Abbas, Yu Yang, Newsha Ardalani, Hugh Leather, and Ari S. Morcos. Sieve: Multimodal dataset pruning using image captioning models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22423–22432, June 2024.

Max Marion, Ahmet Ustun, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When less is more: Investigating data pruning for pretraining LLMs at scale. *arXiv preprint arXiv:2309.04564*, 2023. URL https://arxiv.org/abs/2309.04564.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, 2022. URL https://aclanthology.org/2022.findings-acl.177/.

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15630–15649. PMLR, 2022.

Qwen Team. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Luis Wiedmann, Orr Zohar, Amir Mahla, Xiaohan Wang, Rui Li, Thibaud Frere, Leandro von Werra, Aritra Roy Gosthipaty, and Andrés Marafioti. Finevision: Open data is all you need. *arXiv preprint arXiv:2510.17269*, 2025. URL https://arxiv.org/abs/2510.17269.

Jinda Xu, Yuhao Song, Daming Wang, Weiwei Zhao, Minghua Chen, Kangliang Chen, and Qinya Li. Quality over quantity: Boosting data efficiency through ensembled multimodal data curation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(20):21761–21769, 2025. doi: 10.1609/aaai.v39i20.35481.

Qian Yang, Shivam Chandhok, Oscar Mañas, Kanishk Jain, Aishwarya Agrawal, and Leonid Sigal. Learning what matters: Prioritized concept learning via relative error-driven sample selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15794–15804, June 2026.

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 (ICCV)*, pp. 155–165, October 2025.

Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Long is more for alignment: A simple but tough-to-beat baseline for instruction fine-tuning. *arXiv preprint arXiv:2402.04833*, 2024. URL https://arxiv.org/abs/2402.04833.

APPENDIX CONTENTS

A Implementation Details for Semantic Compression **14**

A.1 Data Construction and Validity Filtering 14

A.2 Representation and Neighborhood Construction 14

A.3 Pair Specificity and Local Semantic Diversity 15

A.4 Integer Budget Allocation and Selection 15

B Implementation Details for Learning-Guided Supplementary **16**

B.1 Fixed Probe Set and Cross-Fitting 16

B.2 Loss Computation and Cluster-Level Aggregation 16

B.3 Directed and Coverage Budgets 17

B.4 Selection Checks and Round Updates 17

C Experimental Details **18**

C.1 Pre-Training and Mid-Training Setup 18

C.2 Incremental Training 18

C.3 Shared Instruction-Tuning Setup 18

C.4 Performance under Different Data Budgets 18

C.5 Detailed Ablation Results 19

D Visualization of Selected and Remaining Data **20**

D.1 Initial Semantic Selection 20

D.2 Directed Selection and Coverage Selection 22

D.3 Selection Behavior across Complementary-Training Rounds 24

A Implementation Details for Semantic Compression

This section provides the data preprocessing, neighborhood construction, score normalization, and budget-rounding details omitted from Sections 2.1 and 2.2.

A.1 Data Construction and Validity Filtering

We sample 10M image–caption pairs from LLaVA-OneVision-1.5-Mid-Training-85M in proportion to its original data sources. Before embedding extraction, we remove only samples that cannot form valid training inputs, including images that cannot be decoded or have invalid dimensions, empty or malformed captions, and entries with missing image paths, text, or sample identifiers. This step does not use semantic similarity or model loss. Samples that fail these checks are excluded from $\mathcal{D}$, whereas valid samples not selected for $\mathcal{S}_0$ remain in the candidate pool $\mathcal{R}_0$.

For the concept-frequency analysis, we count concept occurrences using the 499K MetaCLIP concept vocabulary and sort concepts by decreasing frequency. For the cluster-geometry analysis, we define the normalized centroid of cluster C_c as

$$\mu_c = \mathrm{Norm}\left(\sum_{i \in C_c} m_i\right). \tag{10}$$

For each cluster, we compute the mean maximum-neighbor similarity and mean centroid distance as

$$\rho_c = \frac{1}{n_c} \sum_{i \in C_c} \max_{j \in \mathcal{N}_K(i)} m_i^\top m_j, \qquad \delta_c = \frac{1}{n_c} \sum_{i \in C_c} \left(1 - m_i^\top \mu_c\right). \tag{11}$$

The 16,384 clusters are sorted by size and grouped into 64 equal-count bins for visualization; Figure 2(b) reports the median and interquartile range of ρ_c within each bin. Figure 2(c) plots n_c against ρ_c for every cluster, connecting the bin medians to show the trend; its Spearman correlation is computed over all clusters. The relative change reported in the main text compares the smallest and largest cluster-size deciles.

A.2 Representation and Neighborhood Construction

We encode images, captions, and complete image–caption pairs with the same multimodal embedding model and apply L_2 normalization before all subsequent computations. Consequently, cosine similarity is equivalent to the vector inner product. We apply K-means to the joint representations $\{m_i\}$ with $C = 16{,}384$ clusters. After clustering, we retrieve the $K = 32$ nearest neighbors of every joint representation over the complete dataset, excluding the query itself. This global search is not restricted by cluster boundaries and can therefore identify locally similar pairs that fall on opposite sides of a K-means partition.

Table 4: Hyperparameters used for semantic compression.

Configuration	Value
Candidate data size	10M
Embedding model	Jina-embeddings-v5-omni-small
Embedding dimension	1,024
Clustering algorithm	K-means
Number of clusters C	16,384
Global neighbors K	32
Hard negatives per direction k	10
Semantic-value weight λ	0.5
Cluster-allocation exponent γ	0.7

A.3 Pair Specificity and Local Semantic Diversity

For sample i, let c_i denote its assigned cluster and set $k_i = \min(k, |C_{c_i}| - 1)$. From $C_{c_i} \setminus \{i\}$, we select the k_i captions most similar to image v_i to form $\mathcal{H}_i^T$, and the k_i images most similar to caption t_i to form $\mathcal{H}_i^I$. The bidirectional hard-negative similarities are

$$h_i^T = \frac{1}{k_i} \sum_{j \in \mathcal{H}_i^T} v_i^\top t_j, \qquad h_i^I = \frac{1}{k_i} \sum_{j \in \mathcal{H}_i^I} v_j^\top t_i. \tag{12}$$

We then compute pair specificity as $q_i = s_{ii} - (h_i^T + h_i^I)/2$. A cluster containing only one valid sample has no hard negative; for this case, we set $q_i = s_{ii}$.

To account for the different scales and distributions of q_i and d_i, we use rank-based percentile normalization:

$$Q_i = \frac{\mathrm{rank}_{\mathcal{D}}(q_i) - 1}{N - 1}, \qquad D_i = \frac{\mathrm{rank}_{C_{c_i}}(d_i) - 1}{|C_{c_i}| - 1}. \tag{13}$$

Ties receive their average rank. We normalize q_i globally so that cross-modal matching follows a consistent criterion across semantic regions, whereas d_i is normalized within each cluster to reduce the effect of their intrinsically different densities. We set $D_i = 1$ for singleton clusters.

A.4 Integer Budget Allocation and Selection

Given the real-valued allocation $\widetilde{b}_c^0$, we initialize $b_c^0 = \min(n_c, \lfloor \widetilde{b}_c^0 \rfloor)$ and distribute the remaining budget in descending order of the fractional remainder $\widetilde{b}_c^0 - b_c^0$. Clusters that reach their capacity are removed from further allocation, and the procedure continues until

$$0 \le b_c^0 \le n_c, \qquad \sum_c b_c^0 = B_0. \tag{14}$$

Within each cluster, we retain the b_c^0 samples with the highest S_i^{sem}. Ties are resolved deterministically using q_i, then d_i, and finally the sample identifier. The union of the selected samples forms $\mathcal{S}_0$, while all remaining valid samples form $\mathcal{R}_0 = \mathcal{D} \setminus \mathcal{S}_0$. Samples in $\mathcal{R}_0$ are not classified as low quality at this stage and are reconsidered using model learning states in the subsequent closed loop.

B Implementation Details for Learning-Guided Supplementary

This section supplements Section 2.3 with details of probe construction, cross-fitted reference losses, cluster-level aggregation, and sample selection. Semantic clusters and semantic scores remain fixed across expansion rounds, while target-model losses are recomputed before each round. Target-model training settings are provided in Appendix C.

B.1 Fixed Probe Set and Cross-Fitting

We construct a fixed probe set $\mathcal{P}$ containing 500K samples from the full candidate pool. Following the main-text procedure, we allocate the probe budget evenly across semantic clusters and use pair-specificity scores to guide within-cluster sampling. The probe set is constructed once and reused across all rounds. Probes and their near-duplicates are excluded from both initial and supplementary target-model training data.

For two-fold cross-fitting, we first sort probes stably by sample identifier, then shuffle them within each cluster using a fixed random seed and split them as evenly as possible into folds A and B. For odd-sized clusters, the extra sample is assigned alternately to the two folds, so their sample counts differ by at most one within each cluster. Both probe sampling and fold construction use seed 20260918.

Two reference models are initialized from the same base checkpoint and adapted independently with LoRA on folds A and B. The model adapted on fold A computes reference losses only for fold B, and vice versa. We merge these out-of-fold results by sample identifier to obtain ℓ_i^{ref}. The reference models and reference losses remain fixed in subsequent rounds.

Table 5: Reference-model adaptation settings. The two folds are trained independently with identical hyperparameters.

Configuration	Value
Adaptation method	LoRA
Target modules	Language-model linear layers
Rank / alpha / dropout	16 / 32 / 0.05
Vision encoder / visual merger	Frozen / frozen
Global batch size / micro-batch size	4,096 / 64
Peak / minimum learning rate	2×10^{-4} / 2×10^{-5}
Learning-rate schedule	Cosine decay
Warmup ratio	3%
Maximum sequence length / image tokens	8,192 / 800
Training seed / data seed	20260918 / 20260918

B.2 Loss Computation and Cluster-Level Aggregation

Current and reference losses are computed with teacher forcing and the same supervision mask. After the causal shift, we accumulate negative log-likelihood only over supervised text positions, excluding prompt and padding positions, and divide by the number of valid target tokens in each sample. Samples with no valid target tokens or non-finite losses are excluded from the corresponding statistics.

Cluster-level aggregation weights samples equally, so longer descriptions do not receive greater weight merely because they contain more tokens. To compute $U_c^{(t)}$, we use probes with valid current losses. To compute $R_c^{(t)}$, we first align current and out-of-fold reference losses by sample identifier, then average their differences over probes for which both losses are valid. Each statistic uses its own valid-sample count as the denominator. When all probe losses are valid, this procedure reduces to Eq. 7.

The first round uses the initial mid-training checkpoint θ_0 for diagnosis. Each subsequent round evaluates the same probe set with the preceding round's updated model θ_{t-1} and recomputes $U_c^{(t)}$ and $R_c^{(t)}$. Probe membership, fold assignments, and reference losses remain unchanged.

B.3 Directed and Coverage Budgets

Each round has a budget of $\Delta B_t = 1\text{M}$ samples, with 500K allocated to diagnosis-guided selection. Before selection, we exclude probes, the initial training set, all previous supplementary sets, and their near-duplicates from the remaining candidates.

We set the difficulty-gating fraction to $\rho = 30\%$. Clusters with valid current losses are ranked by decreasing $U_c^{(t)}$; we retain the top 30% and then remove clusters with no available candidates to obtain $\mathcal{G}_t$. The fraction is measured by cluster count. Within $\mathcal{G}_t$, we convert $R_c^{(t)}$ into percentiles $p_c^{(t)}$ and allocate the directed budget using the weights $w_c^{(t)} = 1 + p_c^{(t)}$ defined in Eq. 8. Integer rounding and capacity constraints follow Appendix A.4. Within each cluster, candidates are ranked by the fixed semantic score S_i^{sem} to form $\mathcal{A}_t^{\text{dir}}$.

After directed selection, we remove the selected samples and their near-duplicates from the candidate pool. The coverage budget is

$$B_{\text{cov}}^{(t)} = \Delta B_t - \left|\mathcal{A}_t^{\text{dir}}\right|. \tag{15}$$

We then apply semantic compression to the updated pool to obtain $\mathcal{A}_t^{\text{cov}}$. Cluster-size allocation uses the remaining available sample counts, while cluster assignments and semantic scores remain fixed. Any directed budget left unused because of capacity constraints is thus transferred to coverage selection.

B.4 Selection Checks and Round Updates

The two sets are constructed sequentially, with directed selection preceding coverage selection. Exclusions are enforced using both sample identifiers and near-duplicate groups. After each round, we check overlap between the directed and coverage sets, between the supplementary set and previous training data, and between the supplementary set and the probe set.

The supplementary training set is

$$\mathcal{A}_t = \mathcal{A}_t^{\text{dir}} \cup \mathcal{A}_t^{\text{cov}}. \tag{16}$$

The actual round budget is measured by the size of this union. Total training-data size is measured by the union of $\mathcal{S}_0$ and all supplementary sets. After training on $\mathcal{A}_t$, the updated target model evaluates the fixed probe set to guide the next selection round.

Reference-model adaptation and out-of-fold reference-loss computation are performed once. Target-model probe evaluation, cluster-level aggregation, and supplementary selection are repeated each round. These computations are accounted for as selection overhead separately from target-model training.

C Experimental Details

C.1 Pre-Training and Mid-Training Setup

We initialize the Qwen3-VL-4B architecture with weights from Qwen3-4B and SigLIP2, while randomly initializing the merger. During the initial pre-training stage, we train only the merger on LLaVA-558K. The resulting checkpoint provides the common starting point for all mid-training experiments. Mid-training updates all model parameters, with a global batch size of 4,096, a micro-batch size of 64, a maximum sequence length of 8,192, an image-token limit of 800, and a maximum image pixel count of 819,200. We use full activation recomputation. The learning rate follows a cosine schedule from 10^{-5} to 10^{-6}, with a warmup ratio of 5%. Training on the initial 3M samples runs for one epoch, totaling 732 steps; the one-shot 5M setting runs for 1,220 steps. Training is implemented in Megatron and conducted on 32 NVIDIA A800 GPUs.

C.2 Incremental Training

We perform two incremental training rounds, each using 1M newly selected samples without replaying earlier training data. Each round restores the model weights, optimizer state, random-number-generator state, and cumulative iteration count from the preceding checkpoint and continues updating all model parameters. We use a global batch size of 4,096 and a micro-batch size of 64. The maximum sequence length, image input limits, and activation recomputation settings are shared with initial mid-training. The training seed is 20260918.

The learning-rate schedule is based on the original one-shot 5M mid-training schedule. Initial training on 3M samples follows the first part of this schedule, with warmup set to 5% of the full schedule; subsequent rounds use no additional warmup. The learning rate decays from approximately 4.39×10^{-6} to 1.95×10^{-6} in the first round and from 1.95×10^{-6} to 1.00×10^{-6} in the second. These values are abbreviated for presentation; the training configuration retains the original precision.

The implementation uses piecewise cosine decay with endpoints aligned to the original 5M mid-training schedule, maintaining learning-rate continuity across stage boundaries. The schedules agree at these boundaries, with a small approximation error within each incremental stage.

C.3 Shared Instruction-Tuning Setup

All methods undergo supervised instruction tuning on the same category-balanced subset of 3M FineVision samples. We use a global batch size of 4,096 and a micro-batch size of 32. The learning rate decays from 5×10^{-6} to 5×10^{-7}, with a warmup ratio of 3%. Training runs for one epoch, totaling 732 steps, with a random seed of 20260722. This setup is shared by the one-shot selection baselines and all incremental methods.

C.4 Performance under Different Data Budgets

Table 6: Detailed benchmark results under different data budgets, with 10M samples defined as the 100% budget. Hallusion denotes HallusionBench. All scores are shown to one decimal place. Bold and underlined values indicate the best and second-best distinct results before rounding, respectively.

Method	General				MME		Robustness		TextRich	
	MMBench_test	MMStar	OCRBench	AI2D_test	MME-P	MME-R	Hallusion	POPE	TextVQA	ChartQA
100% budget (10M samples)										
Random	<u>68.9</u>	<u>45.9</u>	<u>59.9</u>	70.6	1492.6	299.3	59.1	**88.1**	**65.8**	<u>64.4</u>
Ours	**70.2**	**47.1**	**60.7**	**71.8**	<u>1526.7</u>	301.1	**63.6**	<u>88.1</u>	<u>63.7</u>	**67.8**
50% budget (5M samples)										
Random	67.3	44.3	56.0	70.0	1500.2	324.3	57.1	87.0	62.0	62.9
Ours	68.5	45.3	57.2	<u>70.7</u>	**1545.5**	**340.4**	57.3	86.3	63.2	63.2
30% budget (3M samples)										
Ours	67.2	43.7	54.6	69.8	1516.7	<u>328.6</u>	<u>60.8</u>	86.1	62.2	62.0

C.5 Detailed Ablation Results

Tables 7, 8, and 9 report the benchmark-level results corresponding to Figure 5, Table 2, and Table 3, respectively. The tables expand the aggregated General, Robustness, and TextRich scores into their constituent benchmarks.

Table 7: Detailed benchmark results for the acquisition and cluster-level learning-diagnosis ablation in Figure 5. Hallusion. denotes HallusionBench. All scores are shown to one decimal place. Bold and underline indicate the best and second-best distinct results before rounding.

Experiment	General				MME		Robustness		TextRich	
	$\text{MMBench}_{\text{test}}$	MMStar	OCRBench	AI2D_test	MME-P	MME-R	Hallusion.	POPE	TextVQA	ChartQA
Acquisition strategy										
A: Random	68.0	44.5	55.6	70.6	1480.0	322.9	54.1	<u>87.7</u>	<u>62.3</u>	63.4
B: Coverage-preserving	<u>68.2</u>	44.9	<u>56.6</u>	70.4	1471.7	<u>327.1</u>	<u>58.7</u>	87.4	61.3	62.7
C: Diagnosis-guided	66.9	<u>45.5</u>	55.2	69.8	**1506.9**	310.7	56.2	86.9	57.1	63.0
Learning-diagnosis signals										
D: Prediction loss	67.9	45.0	55.7	**71.1**	1475.4	319.3	56.8	87.3	56.8	62.9
E: Reference gap	68.0	44.9	56.1	<u>71.0</u>	1431.7	308.2	56.5	**87.9**	60.5	<u>63.5</u>
F: Loss + reference gap	**68.3**	**45.7**	**57.4**	70.8	<u>1480.4</u>	**341.1**	**60.1**	87.4	**64.0**	**64.0**

Table 8: Detailed benchmark results for the semantic supervision scoring ablation in Table 2. Q and D denote pair specificity and local semantic diversity, respectively. Hallusion. denotes HallusionBench. All scores are shown to one decimal place. Bold and underline indicate the best and second-best distinct results before rounding.

Semantic scoring	General				MME		Robustness		TextRich	
	$\text{MMBench}_{\text{test}}$	MMStar	OCRBench	AI2D_test	MME-P	MME-R	Hallusion.	POPE	TextVQA	ChartQA
S1: Random	67.3	44.3	56.0	70.0	<u>1500.2</u>	324.3	<u>57.1</u>	87.4	<u>62.0</u>	62.9
S2: Q only	67.5	<u>44.4</u>	<u>56.9</u>	69.9	1469.7	335.7	54.6	**88.3**	56.4	<u>63.6</u>
S3: D only	<u>67.6</u>	43.7	56.1	<u>70.4</u>	1495.5	<u>340.2</u>	57.1	<u>87.5</u>	59.1	**64.4**
S4: Q+D	**68.5**	**45.3**	**57.2**	**70.7**	**1545.5**	**340.4**	**57.3**	86.3	**63.2**	63.2

Table 9: Detailed benchmark results for the cluster-level budget allocation ablation in Table 3. All settings use the combined Q+D semantic score. Global selection ranks all candidates without cluster quotas; size-proportional and coverage-preserving allocation use $b_c \propto n_c$ and $b_c \propto n_c^{0.7}$, respectively. Hallusion. denotes HallusionBench. All scores are shown to one decimal place. Bold and underline indicate the best and second-best distinct results before rounding.

Budget allocation	General				MME		Robustness		TextRich	
	$\text{MMBench}_{\text{test}}$	MMStar	OCRBench	AI2D_test	MME-P	MME-R	Hallusion.	POPE	TextVQA	ChartQA
Global selection	<u>68.4</u>	43.6	<u>57.1</u>	<u>70.2</u>	1500.5	<u>337.9</u>	57.1	**86.8**	<u>60.1</u>	**63.4**
Size-proportional	68.2	<u>45.3</u>	56.3	<u>70.2</u>	<u>1541.8</u>	333.9	**57.4**	85.4	59.4	62.6
Coverage-preserving	**68.5**	**45.3**	**57.2**	**70.7**	**1545.5**	**340.4**	<u>57.3</u>	<u>86.3</u>	**63.2**	<u>63.2</u>

D VISUALIZATION OF SELECTED AND REMAINING DATA

This section visualizes representative image–text pairs from the original LLaVA-OneVision 20M candidate pool. The images and shortened source descriptions are taken from the data records. Colored text marks the phrases that most clearly distinguish two otherwise related samples. Ellipses indicate omitted portions of the original descriptions.

D.1 INITIAL SEMANTIC SELECTION

The initial selector favors examples that jointly provide reliable image–text correspondence, semantic distinctiveness, and coverage of the candidate pool. The following pairs illustrate why two samples from a broadly similar topic can receive different selection outcomes.

Pair 1: Fine-grained descriptions of earrings. Both records depict earrings, but the selected example provides a denser description of the local materials, colors, and part structure. This makes it more informative for learning fine-grained visual grounding.



Selected
$Q = 0.84,\ D = 0.62,\ S_{\text{sem}} = 0.73$
Source-text : "…a decorative arrangement of **small pink pearls clustered around a larger, teardrop-shaped lavender pearl**. The gold elements include **a ribbed cylindrical bead between the small and large pearls** …"



Remaining
$Q = 0.61,\ D = 0.34,\ S_{\text{sem}} = 0.475$
Source-text : "…Each earring has an elongated oval shape with a smaller connecting link; the left earring has **a solid gold large hoop and a smaller diamond-encrusted link**, while the right earring reverses this arrangement …"

Figure 6: **Initial selection—earrings.** The selected record verbalizes the color, material, geometry, and spatial composition of the jewelry at a finer granularity. Scores are illustrative.

Pair 2: Two wristwatches with different descriptive precision. The examples share the same coarse semantic category. The selected record, however, explicitly grounds several visible components—band material, dial color, hour markers, hands, and date window—while the remaining record provides fewer mutually reinforcing details.



Selected
$Q = 0.84,\ D = 0.60,\ S_{\text{sem}} = 0.72$
Source-text : "... **silver stainless-steel mesh band** ... The round watch face is **deep blue with white hour markers and hands, including a date window at the 3 o'clock position** ..."



Remaining
$Q = 0.60,\ D = 0.52,\ S_{\text{sem}} = 0.56$
Source-text : "... a **white dial with thin gold-tone hour markers and hands** ... The case is rose-gold colored, complementing the **black woven nylon strap** ..."

Figure 7: **Initial selection—wristwatches.** The pair controls for the coarse object category and exposes the effect of description specificity. Scores are illustrative.

Pair 3: Natural scenes with different coverage value. Both samples are visually valid outdoor scenes. The selected image combines several grounded elements—flowers, leaves, tree trunks, and their spatial arrangement—whereas the remaining image is dominated by a repeated grass texture that is already common in the pool.



Selected
$Q = 0.80,\ D = 0.74,\ S_{\text{sem}} = 0.77$
Source-text : "A vast field is carpeted with **small, bright purple flowers against lush green leaves** ... **Two thick, brown tree trunks rise from the middle of the field** ..."



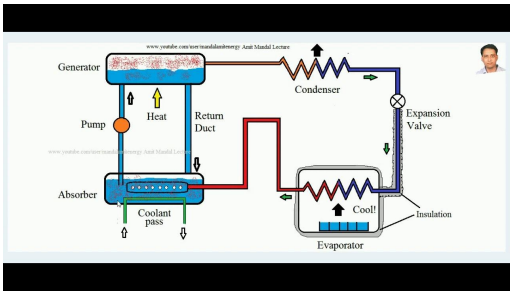
Remaining
$Q = 0.76,\ D = 0.18,\ S_{\text{sem}} = 0.47$
Source-text : "... **tall, slender green grass or wheat stalks with soft, feathery tips** ... the background is a blurred sea of deeper green ..."

Figure 8: **Initial selection—natural scenes.** Comparable image–text quality does not imply equal coverage value: the selected example contributes a less repetitive semantic composition. Scores are illustrative.

D.2 Directed Selection and Coverage Selection

Each complementary-training round combines two mutually exclusive subsets. Directed selection allocates samples to semantic clusters in which the current model exhibits high uncertainty and a large cross-fitted learning gap. Coverage selection then fills the unspent budget from the remaining pool using the semantic-compression objective. The examples below illustrate the difference between these two routes. A category is not intrinsically "directed" or "coverage"; the assignment depends on the round-specific cluster statistics and the available candidates.

Pair 1: A rare technical diagram and a common food photograph. The absorption-refrigeration diagram contains a structured system of components and directional relations. Under the illustrative diagnostics, its cluster has high current loss and a substantial learning gap, so it enters the directed subset. The food photograph is well aligned and visually informative, but its cluster is already easier for the model; it is retained through coverage selection.



Directed selection
$U = 2.10,\ R = 0.55,\ Q = 0.79,\ D = 0.58,\ S_{\text{sem}} = 0.685$
Source-text : "…labeled components including a **Generator, Pump, Absorber, Condenser, Expansion Valve, and Evaporator**. **Colored arrows indicate fluid or heat-flow directions** …"



Coverage selection
$U = 1.20,\ R = 0.16,\ Q = 0.82,\ D = 0.76,\ S_{\text{sem}} = 0.79$
Source-text : "**Four bacon-wrapped pasta cups** sit on a white plate …A fork pierces one cup, **stretching melted cheese** …"

Figure 9: **Directed versus coverage selection—system diagram and food scene.** The technical diagram is selected to address a diagnosed weakness; the food sample broadens the retained visual distribution. Diagnostics and scores are illustrative.

Pair 2: Tabular scientific content and an everyday interior. The scientific table requires reading headers, aligning rows and columns, and associating numerical values with chemical abbreviations. The sofa image is a more common everyday scene, yet it preserves useful variation in material, color, and decorative details.

	Total lipid content (g/100g)	EPA (% of total fatty acids)	DHA (% of fatty acids)
Ling	51-66	5.1-7.7	10.2-12.9
Saithe	48-76	4.4-9.0	10.5-13.3
Hake	42-67	5.5-7.7	11.7-13.0
Cod	50-64	6.0-8.6	12.3-14.4
Monkfish	31-43	4.6-6.3	10.8-13.1

Table 2. Variation in content of lipids, EPA and DHA in livers from cod fish during 21 months

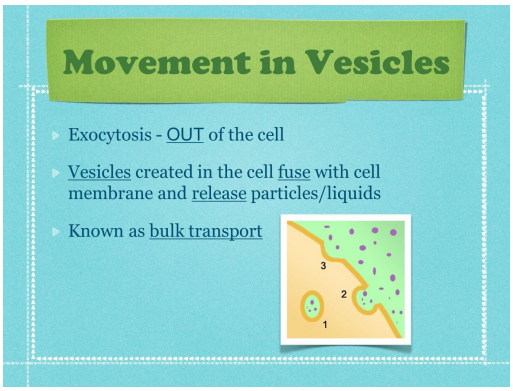
Directed selection
$U = 2.35,\ R = 0.65,\ Q = 0.85,\ D = 0.55,\ S_{\text{sem}} = 0.7$
Source-text : "...a table with columns for **Total lipid content, EPA, and DHA**, and rows for fish species including **Ling** ..."



Coverage selection
$U = 1.12,\ R = 0.12,\ Q = 0.84,\ D = 0.66,\ S_{\text{sem}} = 0.75$
Source-text : "...a **dark-blue leather loveseat** with **gold-and-blue patterned throw pillows**, rolled arms, and small brass feet ..."

Figure 10: **Directed versus coverage selection—scientific table and sofa.** Directed selection concentrates budget on a cluster with a large diagnosed learning deficit, while coverage selection retains a representative everyday scene. Diagnostics and scores are illustrative.

Pair 3: A biological process and a residential exterior. The instructional slide encodes a temporally ordered process and requires joint interpretation of arrows, labels, and schematic objects. The house image is visually conventional but provides complementary coverage of architecture and spatial layout.



Directed selection
$U = 2.05,\ R = 0.45,\ Q = 0.83,\ D = 0.63,\ S_{\text{sem}} = 0.73$
Source-text : "...a vesicle **moves toward the cell membrane, fuses with it, and releases its contents outside the cell** ..."



Coverage selection
$U = 1.15,\ R = 0.10,\ Q = 0.81,\ D = 0.59,\ S_{\text{sem}} = 0.7$
Source-text : "...a house with a **wraparound porch, white railings, shutters, and a dormer**, approached by a curved path ..."

Figure 11: **Directed versus coverage selection—biological process and house exterior.** The first sample addresses a process-understanding weakness, whereas the second protects broad scene coverage. Diagnostics and scores are illustrative.

D.3 Selection Behavior across Complementary-Training Rounds

The probe split, cross-fitted reference loss, semantic clusters, and semantic scores remain fixed across rounds. Before each new round, we recompute the current-model loss and hence the cluster-level uncertainty U and learning gap R. Selection can therefore change because the target model changes, because the percentile ranking of a cluster changes, or because earlier selections alter the available candidate pool. The following cases make these mechanisms concrete.

Case 1: A chart enters the directed subset in the second round. The absolute learning gap decreases after the first complementary-training round, but other clusters improve more strongly. The chart cluster therefore moves upward in the round-wise percentile ranking and receives directed budget in Round 2.



$Q = 0.66,\ D = 0.61,\ S_{\text{sem}} = 0.635$

Fixed cross-fitted reference loss: 1.60

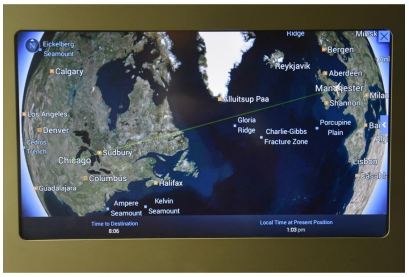
Round 1: $U = 1.80,\ R = 0.20$
Outcome: not selected; the cluster falls below the directed-budget cutoff.

⇓

Round 2: $U = 1.70,\ R = 0.10$
Outcome: selected by the directed branch; its relative rank rises after the model improves more on competing clusters.

Figure 12: **Across-round case 1—relative ranking.** A smaller absolute gap can still receive budget when the round-wise distribution of diagnostic signals changes. Values and outcomes are illustrative.

Case 2: A route map is admitted by the coverage branch. The route-map cluster remains outside the directed gate in both rounds. After Round 1 removes other high-scoring candidates and their near duplicates, the map becomes competitive under the updated coverage allocation and is selected in Round 2.



$Q = 0.82,\ D = 0.62,\ S_{\text{sem}} = 0.72$

Fixed cross-fitted reference loss: 1.25

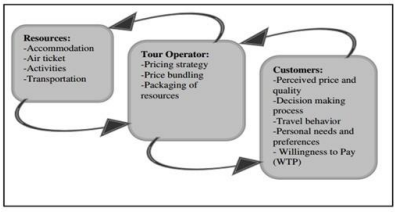
Round 1: $U = 1.50,\ R = 0.25$
Outcome: not selected; no directed quota is assigned and the sample is below the coverage cutoff.

⇓

Round 2: $U = 1.35,\ R = 0.10$
Outcome: selected by the coverage branch; the available candidate counts and coverage cutoff change after Round 1.

Figure 13: **Across-round case 2—candidate-pool update.** Coverage selection is recomputed on the remaining pool, so a sample can become useful even without entering the difficulty gate. Values and outcomes are illustrative.

Case 3: A selected process diagram is excluded from later rounds. The flowchart is selected by the directed branch in Round 1. It is then removed from the candidate pool together with its near duplicates. Its improved Round 2 diagnostics can still be measured on the fixed probe, but the training sample itself is ineligible for reselection.



$Q = 0.83,\ D = 0.60,\ S_{\text{sem}} = 0.715$

Fixed cross-fitted reference loss: 1.50

Round 1: $U = 2.00,\ R = 0.50$
Outcome: selected by the directed branch; the cluster passes the difficulty gate and receives quota.

⇓

Round 2: $U = 1.65,\ R = 0.15$
Outcome: excluded from the candidate pool; the sample and its near duplicates were consumed in Round 1.

Figure 14: **Across-round case 3—history-based exclusion.** The selection sets remain disjoint because previously used samples and their near duplicates are removed before the next round. Values and outcomes are illustrative.