DATAVALVE: SELECTING-WHILE-TRAINING VIA UTILITY-GATED ROUTING FOR VISUAL INSTRUCTION TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual instruction tuning has become increasingly expensive as multimodal large language models (MLLMs) and training corpora scale. Existing data selection methods mostly follow an offline *select-then-train* pipeline: they score a large candidate pool, often with an additional target MLLM traversal, and then fine-tune on a fixed subset. This pipeline is both costly and blind to the evolving training state, so it cannot avoid samples whose marginal utility has already decayed. We propose DATAVALVE, a *selecting-while-training* framework for fixed-pool visual instruction tuning. A lightweight router selects training samples from pre-extracted features and adapts through delayed golden-set loss feedback. To counter the tendency of loss-only feedback to repeatedly favor high-loss sources, an online LoRA-influence gate modulates credit using relative update strength. This coupling prioritizes currently useful samples and suppresses redundant low-impact updates without external teachers, downstream validation sets, or explicit diversity losses. Experiments on LLaVA-v1.5-7B show that DATAVALVE routes only 20% of training data through target MLLM training while achieving **98.6%** of full-data fine-tuning performance, outperforming strong baselines under the same routing budget with the best efficiency–accuracy balance. Further experiments demonstrate transfer across candidate pools, model scales, and MLLM families. Our code is included in the supplementary material.

1 INTRODUCTION

Instruction tuning has become fundamental for aligning multimodal large language models (MLLMs) with human intent, enabling recent MLLMs Gemini Team, Google (2024); Bai et al. (2025); Liu et al. (2023) to reason over visual and textual inputs for tasks including visual question answering Antol et al. (2015); Hao et al. (2024); Wu et al. (2017), multimodal reasoning Lu et al. (2024); Yue et al. (2024); Lu et al. (2022); Zheng et al. (2023), and visual grounding Deng et al. (2021); Wang et al. (2024b); Yan et al. (2024). The growing scale and substantial redundancy of visual instruction datasets pose a practical challenge: how can we reduce MLLM training computation without sacrificing downstream performance?

Data selection for MLLM fine-tuning is challenging because visual instruction datasets mix heterogeneous image-text pairs and instruction formats with widely varying difficulty and redundancy. Thus, sample value cannot be judged by surface-level quality alone. Beyond differences across examples, **usefulness changes as the model learns**: a once-informative sample may become redundant, while a simple, rare sample can become valuable after common patterns are absorbed.

However, existing visual instruction data selection methods largely follow an *offline select-then-train* paradigm Liu et al. (2025); Wu et al. (2024); Chen et al. (2024); Yan et al. (2025): they score candidates before fine-tuning the target MLLM on a fixed subset. Figure 1 summarizes two limitations. **For selection performance**, Figure 1(a, top) illustrates how an offline scorer may favor detailed descriptions of common tigers over a simpler example of a rarer white tiger. Scores fixed before target training cannot reflect that common patterns may already be learned while the rare concept still offers higher marginal utility to the current model. **For selection efficiency**, Figure 1(b, top) shows that scoring and fine-tuning are separate stages. Methods that score the full pool with the

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.

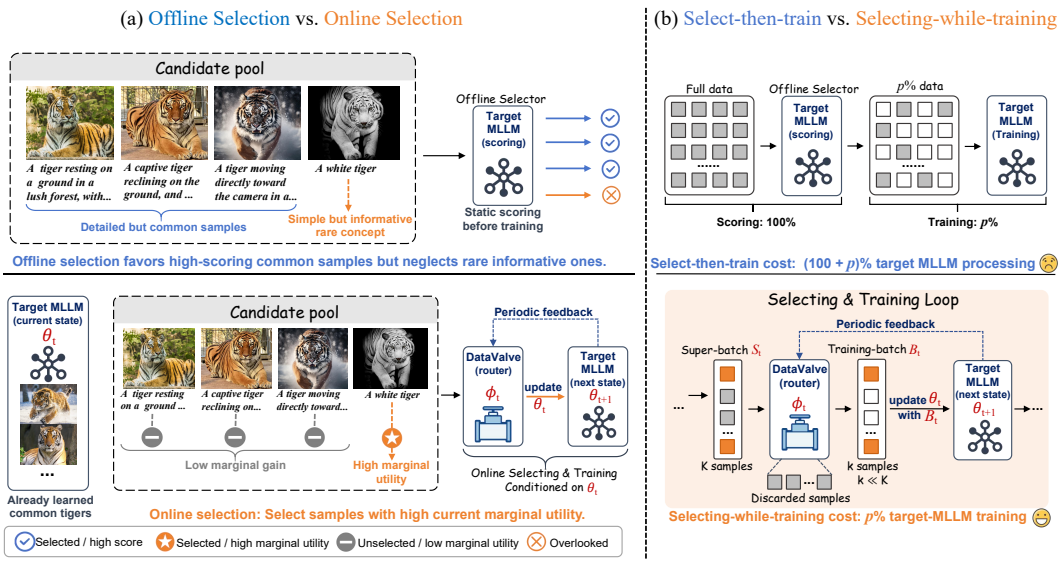


Figure 1: **Offline selection versus DATAVALVE. (a)** Offline selection fixes sample scores before final tuning, whereas DATAVALVE performs state-dependent routing for high current marginal utility. **(b)** Offline methods that score the full pool with the target MLLM incur an additional model traversal before subset fine-tuning, whereas DATAVALVE avoids this separate full-pool scoring pass by integrating routing with training.

target MLLM process 100% of candidates for scoring before training on the selected $p\%$, yielding $(100 + p)\%$ example-level target-MLLM scoring/training passes across the two stages.

Motivated by these limitations, we propose **DATAVALVE**, a novel *selecting-while-training* framework for visual instruction tuning. Instead of assigning static scores before fine-tuning, DATAVALVE treats data selection as a state-aware routing problem: a lightweight router acts as a data valve, dynamically deciding which samples should enter target MLLM training according to the evolving training state.

This design addresses both limitations. As Figure 1(a, bottom) illustrates, DATAVALVE conditions routing on the current training state, avoiding samples whose value has decayed after common patterns are learned and prioritizing those with higher current marginal utility, such as rare visual concepts not yet absorbed by the model. Figure 1(b, bottom) shows how selection is integrated into training: the router chooses a small training batch from each candidate ***super-batch***, and only routed samples receive target MLLM forward/backward updates. Thus, dominant target MLLM computation is spent on the selected $p\%$ training samples without a separate full-pool target-model scoring pass.

The central challenge is to obtain a low-cost learning signal that tracks training progress. We reserve a small golden set from the original instruction pool, exclude it from target-model training, and use its loss changes as held-out feedback. However, loss-only rewards can repeatedly favor sources with large immediate loss-reduction potential, even as their broader marginal value declines. We therefore modulate this feedback with an online LoRA-influence gate, reusing training gradients to compare routed-update strength against recent history. The gate reduces credit magnitude for weak repeated updates while preserving feedback sign, supporting adaptive routing without explicit diversity losses. In summary, our contributions are three-fold:

(i) We introduce DATAVALVE, a novel selecting-while-training framework for visual instruction tuning. By integrating selection into target MLLM fine-tuning, DATAVALVE avoids a separate full-pool target-model scoring pass and trains on only the selected samples. **(ii)** We design a utility-gated routing objective that combines golden set loss feedback with an online LoRA-influence gate. This enables the router to down-weight redundant batches and up-weight high-marginal-utility samples without external teachers, downstream validation sets, or explicit diversity optimization losses. **(iii)** Experiments across nine benchmarks show that DATAVALVE achieves 98.6% of full-data fine-tuning performance with a strong performance–efficiency trade-off.

2 Related Work

Visual Instruction Data Selection. Visual instruction data selection typically constructs a compact subset before training. Prior work uses VLM-based filtering Wei et al. (2023), loss-based selection Chen et al. (2024), gradients Liu et al. (2025); Wu et al. (2024), transferability Lee et al. (2024), multimodal representations Bi et al. (2025), scalable filtering Wu et al. (2026), and pre-instruction data construction Safaei et al. (2025). Although these approaches improve the quality of selected subsets, the selection process itself can introduce substantial computational overhead on large multimodal instruction pools. Recent methods therefore focus on efficient selection. CoIDO Yan et al. (2025) trains a lightweight scorer on a small subset, while VisNec Dong et al. (2026) measures visual necessity with paired target-model forwards but no backward scoring. However, both retain a select-then-train paradigm: their subsets are fixed before final target-MLLM fine-tuning.

Dynamic Data Allocation During Training. Adapting data use to the evolving model has been explored in active learning, curriculum and self-paced learning, and online hard example mining Settles (2009); Bengio et al. (2009); Kumar et al. (2010); Shrivastava et al. (2016). Recent methods include holdout-loss prioritization Mindermann et al. (2022), online batch selection Wang et al. (2024a), and iterative instruction selection Song et al. (2024); Lin et al. (2025). In multimodal learning, OASIS Lee et al. (2025) and Adapt-∞ Maharana et al. (2025) study continual or lifelong data streams. Visual instruction tuning, by contrast, typically starts from a fixed instruction pool: the question is how to allocate limited target-MLLM training computation among existing examples as the model evolves. DataValve studies this setting through utility-gated routing and delayed target-MLLM feedback, without a separate full-pool target-MLLM scoring stage.

3 Method

Task Description. Given a visual instruction tuning dataset $\mathcal{D} = \{z_j\}_{j=1}^{N}$, where each sample $z_j = (I_j, x_j, y_j)$ contains an image, an instruction, and a response, our goal is to train a target MLLM using only a small fraction of the available training data. Unlike offline selection, which first constructs a fixed subset before fine-tuning, online data selection makes selection decisions during fine-tuning according to the evolving target model state.

Let $p\%$ denote the target selection ratio, and let $\mathcal{B} = \frac{p}{100}N$ denote the total training budget, which is the maximum number of samples allowed to update the target MLLM. At step t, the selector observes a large incoming batch $S_t \subseteq \mathcal{D}$ and chooses a smaller training subset $B_t \subseteq S_t$ for the current target model update. The online selection problem can be written as

$$\max_{\{B_t\}_{t=1}^{T}} \mathrm{Perf}(\theta_T), \qquad \text{s.t.} \quad B_t \subseteq S_t, \ \sum_{t=1}^{T} |B_t| \le \mathcal{B} = \frac{p}{100}N,$$

where θ_T is the final target MLLM and $\mathrm{Perf}(\theta_T)$ denotes downstream evaluation performance.

Golden Set Construction. DataValve constructs a small held-out golden set $\mathcal{G}$ from the original training pool to provide intrinsic feedback without downstream validation sets or external teacher models. Before online training, we sample a small probe subset from $\mathcal{D}$, run the untuned target MLLM in forward mode, sort probed samples by loss, and keep an intermediate percentile range $[\tau_l, \tau_u]$; e.g., $[40, 90]$ removes the easiest 40% and the highest-loss 10%, leaving medium-to-high difficulty samples. We randomly draw $|\mathcal{G}|$ samples from this range as the golden set. This construction is simple and low-overhead: it uses only forward losses on a small probe subset and requires no extra training, annotation, or teacher model.

The golden set is removed from training, and the remaining data form $\mathcal{C} = \mathcal{D} \setminus \mathcal{G}$. Thus, $\mathcal{G}$ is used only for feedback, while $\mathcal{C}$ supplies online super-batches. To reduce validation overhead, we split $\mathcal{G}$ into four rotating shards $\{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4\}$ and evaluate only the active shard $\mathcal{G}_r$ periodically, with r cycling over the four shards.

Overall Framework. Figure 2 illustrates the data flow. Training starts with a short random warm-up using $p_w\%$ of samples, counted toward the total budget, to avoid early router decisions before any policy feedback exists. Because DataValve performs online selection over target MLLM updates, these random warm-up samples are treated as already selected under the same budget. Small random

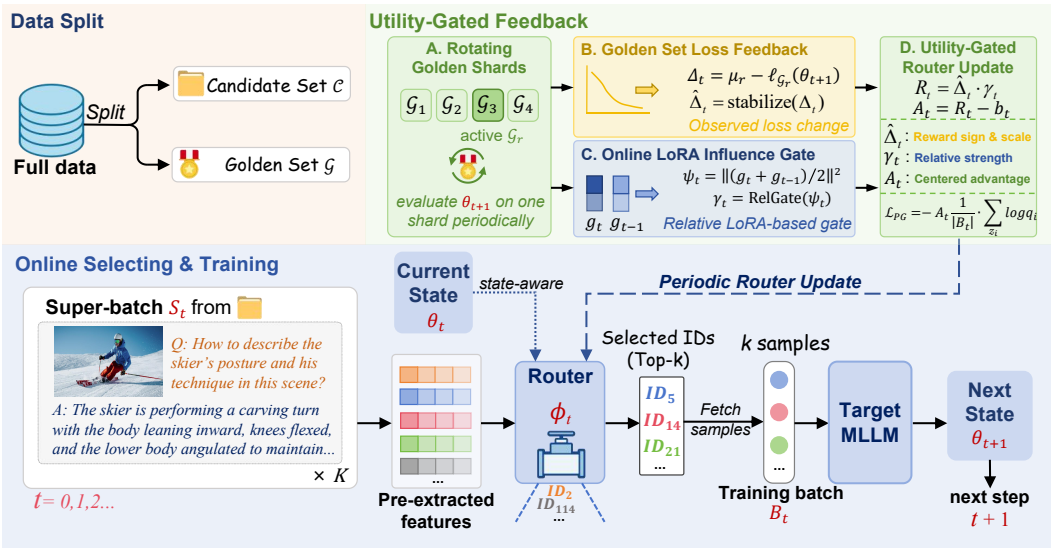


Figure 2: **Framework overview of DATAVALVE.** The router selects target MLLM training batches from candidate super-batches and is periodically updated using rotating golden-shard feedback and an online LoRA influence gate. The one-update schematic uses θ_{t+1}; implemented delayed feedback uses $\theta_{s(t)}$ as specified in Eq. 4 and Algorithm 1.

warm-up ratios, e.g., 3–5%, are unlikely to dominate the final selected distribution and make later routing more stable Xia et al. (2024b). We denote the target MLLM after this warm-up stage by θ_0; router training and feedback then start from this warmed-up state.

After warm-up, each step samples a candidate super-batch S_t of size K from $\mathcal{C}$ and represents candidates with pre-extracted multimodal features, e.g., CLIP features Radford et al. (2021). Let b be the standard MLLM batch size. The router scores all candidates and selects Top-k with $k = b$, so the routed batch matches the ordinary training batch size. To meet the remaining budget, we set the router-stage selection ratio to $\rho_{\text{route}} = (p - p_w)/(100 - p_w)$ and choose K such that $b/K \approx \rho_{\text{route}}$. **$K$ is determined by this budget constraint.**

The router outputs sample IDs rather than materialized tensors. Only selected raw samples are fetched to form the MLLM training batch B_t, and the target MLLM update produces θ_{t+1}. We call the router state-aware because its policy is periodically updated from previous target MLLM state transitions, not because full model parameters are fed as router inputs. The target MLLM and router use separate losses and optimizers: target training updates only trainable target parameters, while the router loss updates only ϕ from stored, detached features and feedback.

Router Training. The router is an online selection policy. For candidate feature e_i of $z_i \in S_t$, it produces a logit $a_i = r_\phi(e_i)$ and probability $q_i = \sigma(a_i)$. Top-k routing defines a binary action $\mathcal{A}_t = \{m_i\}_{z_i \in S_t}$, where $m_i = 1$ means that z_i enters B_t.

After warm-up, periodic golden-shard feedback updates the stored action from the preceding feedback event (Figure 2); the event schedule is specified in Appendix B.

3.1 UTILITY-GATED ROUTER OPTIMIZATION

Influence-Motivated Reward Decomposition. For a single routed batch B_t updating the target MLLM from θ_t to θ_{t+1}, an ideal one-step utility would be the change in held-out golden-shard loss:

$$U_t = \ell_{\mathcal{G}_r}(\theta_t) - \ell_{\mathcal{G}_r}(\theta_{t+1}). \tag{1}$$

Positive U_t means the batch improves the golden shard; negative values hurt the held-out objective.

A first-order influence-function view Koh & Liang (2017); Xia et al. (2024a) clarifies what this utility contains. In our setting, the trainable target parameters are the LoRA parameters, denoted by θ^L. For

the routed batch B_t, the target update is locally given by

$$\theta_{t+1}^L \approx \theta_t^L - \eta_t^L g_t, \qquad g_t = \nabla_{\theta^L} \ell_{\text{train}}(\theta_t; B_t), \tag{2}$$

where η_t^L is the LoRA learning rate. Applying a Taylor expansion to the golden-shard loss gives

$$\ell_{\mathcal{G}_r}(\theta_{t+1}^L) - \ell_{\mathcal{G}_r}(\theta_t^L) \approx -\eta_t^L \nabla_{\theta^L} \ell_{\mathcal{G}_r}(\theta_t)^\top g_t. \tag{3}$$

Thus, the first-order utility term depends on both golden-gradient alignment and routed-update magnitude. Directly computing alignment would require an extra golden-shard backward pass at each feedback event. Instead, observed golden-loss feedback supplies a delayed signed outcome, while existing LoRA training gradients provide a low-cost proxy for relative update strength.

By Cauchy-Schwarz, the magnitude of this first-order term, not the exact U_t, is bounded (up to the learning-rate factor) by the product of the golden-gradient norm and routed-batch LoRA-gradient norm. This makes update strength an action-dependent factor in a potential-utility bound; it does not make gradient magnitude a direct measure of usefulness. The smoothed energy used by the gate below is a practical proxy for that strength (Appendix A.1).

Golden-Set Loss Feedback. Let $\ell_{\mathcal{G}_r}(\theta)$ denote the target MLLM's average loss on the active golden shard $\mathcal{G}_r$. For a stored action from step t, let $s(t) > t$ be its feedback step. Evaluation precedes the optimizer step at $s(t)$, so $\theta_{s(t)}$ includes the update on B_t and intervening updates. Using a recent shard-local loss baseline μ_r, we assign

$$\Delta_t = \mu_r - \ell_{\mathcal{G}_r}(\theta_{s(t)}), \tag{4}$$

Positive Δ_t indicates improvement relative to the baseline. This delayed credit is a practical surrogate for the stored action, not an isolated causal estimate of its one-step utility in Eq. 1. For comparable and stable feedback across rotating shards, the `stabilize` operation in Figure 2 divides Δ_t by a recent shard-specific loss scale with a numerical floor and symmetrically clips extreme magnitudes. We denote the resulting signed signal by $\hat{\Delta}_t$; stabilization preserves its sign while limiting the influence of outlying evaluations.

Online LoRA Influence Gate. Golden-set feedback compares the evaluated model state with a recent shard-local baseline, but it does not measure update strength. Since patterns can lose marginal utility once absorbed, we use routed-batch LoRA-gradient energy as a low-cost update-strength proxy.

A single $\|g_t\|_2^2$ can fluctuate across feedback slots. We therefore average the current and previous captured LoRA gradients; below, g_{t-1} denotes the previous valid feedback-slot gradient:

$$\bar{g}_t = \tfrac{1}{2}(g_t + g_{t-1}), \qquad \psi_t = \|\bar{g}_t\|_2^2. \tag{5}$$

Equivalently,

$$\psi_t = \tfrac{1}{4}\left(\|g_t\|_2^2 + \|g_{t-1}\|_2^2 + 2\|g_t\|_2\|g_{t-1}\|_2 \cos\varphi_t\right), \tag{6}$$

where

$$\cos\varphi_t = g_t^\top g_{t-1} \;/\; \max(\|g_t\|_2\|g_{t-1}\|_2, \epsilon). \tag{7}$$

This statistic is large when the routed batch induces a strong LoRA update consistent with the recent update direction, and small when the update is weak or cancels recent directions. We do not require it to be a calibrated estimate of exact parameter displacement; rather, it is an energy-style proxy for the LoRA update strength induced by the routed batch. We use squared LoRA-gradient energy as a discriminative measure before logarithmic compression.

Absolute gradient energy is poorly calibrated across training stages because LoRA gradient norms naturally decay. To remove drift, we compare the current update strength with a short recent history of valid feedback events. We compute $s_t = \log(1 + \psi_t)$ and apply short-window standardization:

$$\gamma_t = 0.5 + \text{sigmoid}\big(\text{Norm}_H(s_t)\big), \tag{8}$$

where $\text{Norm}_H(\cdot)$ denotes standardization using the mean and variance of the previous H valid feedback events; the explicit z-score form is provided in Appendix A.1. We denote this mapping by $\gamma_t = \text{RelGate}(\psi_t)$. The gate is centered at 1 and lies in $(0.5, 1.5)$, so below-normal LoRA update strength yields weaker credit and above-normal strength receives stronger credit. Since $\gamma_t > 0$, the gate cannot reverse the golden-feedback sign. It uses gradients already produced by target training and appends the current s_t only after computing γ_t, so the batch does not normalize itself.

Utility-Gated Router Update. The utility-gated feedback for the routed action is

$$R_t = \hat{\Delta}_t \cdot \gamma_t. \tag{9}$$

Here, $\hat{\Delta}_t$ provides the reward sign and normalized loss-feedback scale, while γ_t modulates credit strength without changing the sign.

Online target MLLM training is nonstationary: as the target model evolves, the scale and offset of golden feedback can drift across feedback events. To make the router compare actions against nearby feedback rather than a fixed zero-centered threshold, we use a standard advantage baseline:

$$b_t = \frac{1}{H} \sum_{j=1}^{H} R_{t-j}, \qquad A_t = R_t - b_t. \tag{10}$$

For early feedback events, we use available previous rewards once the minimum history count is reached; before that, $A_t = R_t$. The reward baseline uses the same window length H as the LoRA gate, so all recent-history baselines share a time scale. Since b_t is computed from previous feedback events, it is independent of the current routed action and serves as a standard policy-gradient baseline. Thus, R_t is the utility-gated feedback, while A_t is the centered learning signal. Formal details on the advantage baseline are provided in Appendix A.2.

The router is optimized with a policy-gradient-style surrogate inspired by REINFORCE Williams (1992). Because Top-k routing is discrete, gradients cannot pass through selected IDs directly. We view the router probabilities as a Bernoulli policy

$$\pi_\phi(\mathcal{A}_t | S_t) = \prod_{z_i \in S_t} q_i^{m_i} (1 - q_i)^{1 - m_i}, \tag{11}$$

followed by a fixed-budget Top-k projection. Since the deterministic projection already enforces the budget, we use the selected action log-likelihood as a practical surrogate and omit unselected $\log(1 - q_i)$ terms (see Appendix A.3 for details); lightweight stabilization handles global logit scale.

$$\mathcal{L}_{\mathrm{PG}} = -A_t \cdot \frac{1}{|B_t|} \sum_{z_i \in B_t} \log q_i. \tag{12}$$

When $A_t > 0$, the current utility feedback is better than the recent baseline, so minimizing Eq. 12 increases selected-sample probabilities; when $A_t < 0$, it decreases them. This gives an update-aware path from MLLM improvement to routing, while target MLLM and router gradients remain isolated.

Implicit Redundancy Control. The utility-gated feedback in Eq. 9 also discourages redundant routing without explicit diversity losses. Rotating golden shards provide a broad intrinsic signal, making over-specialized policies less likely to improve consistently across shards. Meanwhile, the LoRA gate weakens credit for actions whose update strength falls below the recent local norm: as repeated exposure saturates the target MLLM on a pattern, LoRA-gradient energy should shrink relative to nearby routed batches, moving γ_t below 1. Thus, redundant actions can receive weaker reinforcement even when golden set feedback remains positive. This encourages diversity indirectly by favoring batches with above-normal update strength that likely retain high current marginal utility.

4 EXPERIMENTS

4.1 SETUP

Basic Setting. We evaluate DATAVALVE on the `LLaVA-665K` visual instruction tuning dataset Liu et al. (2024a), a heterogeneous mixture of visual QA, OCR, captioning, grounding, and conversational data. We use LLaVA-v1.5-7B-LoRA Liu et al. (2023) as the target MLLM. Unless otherwise specified, the target training budget is 20% of the candidate pool, and all target model hyperparameters follow the standard LLaVA LoRA configuration. We reserve $|\mathcal{G}| = 2048$ samples from `LLaVA-665K` as the golden set and remove them from the candidate pool. The 3% random warm-up is counted within the total target training budget. The router is a lightweight four-layer MLP. We set the recent-history window length to $H = 16$ by default and study its sensitivity in Appendix D.1. Full implementation and experimental details are in Appendices B and C.

Table 1: Performance, total FLOPs, and wall-clock time using 20% of `LLaVA-665K` for LLaVA-v1.5-7B-LoRA fine-tuning. †The original LLaVA-v1.5 report used GPT-4-0314, which is no longer available; we follow OpenAI's recommended replacement GPT-4o-0513.

Method	VQAv2	GQA	SQA-I	TextVQA	POPE	MME	MMBench en	MMBench cn	LLaVA-Bench†	Rel. (%)	Total FLOPs (EFLOPs)	Wall-clock Time (h)
Full-Finetune	79.1	63.0	68.4	58.2	86.4	1476.9	66.1	58.9	59.3	100.0	10.2	22.7
Random	74.1	58.6	67.5	55.3	85.7	1461.0	59.8	53.4	56.3	95.0	**2.0**	**4.5**
SELF-FILTER Chen et al. (2024)	73.7	58.3	61.4	52.9	83.8	1306.2	48.8	45.3	57.9	88.9	33.0	75.9
COINCIDE Lee et al. (2024)	76.5	59.8	69.2	55.6	86.1	**1495.6**	63.1	54.5	57.3	97.1	6.8	13.4
TIVE Liu et al. (2025)	76.0	58.4	69.8	53.3	85.7	1448.4	**66.9**	58.7	56.4	97.3	13.6	27.4
ICONS Wu et al. (2024)	77.0	**60.4**	70.4	54.5	86.1	1447.7	64.6	54.0	59.2	97.4	14.4	28.5
COIDO Yan et al. (2025)	**77.2**	**60.4**	69.4	55.6	85.4	1450.2	63.8	56.7	57.8	97.5	6.2	11.6
VISNEC Dong et al. (2026)	76.9	60.2	67.2	55.9	85.1	1443.3	66.0	**59.1**	56.6	97.7	8.0	9.5
Ours	**77.2**	59.7	**70.8**	**56.7**	**87.5**	1454.0	63.8	56.9	**60.2**	**98.6**	2.3	6.5

Evaluation Tasks. Following prior selection work Wu et al. (2024); Lee et al. (2024), we evaluate on benchmarks covering complementary MLLM capabilities: VQAv2 (Goyal et al., 2017) and GQA (Hudson & Manning, 2019) for visual question answering, SQA-I (Lu et al., 2022) for knowledge-grounded reasoning, TextVQA (Singh et al., 2019) for OCR-oriented understanding, POPE (Li et al., 2023) for hallucination, MME (Fu et al., 2023) and MMBench (Liu et al., 2024b) for general multimodal evaluation, and LLaVA-Bench (Liu et al., 2023) for free-form generation.

4.2 OVERALL COMPARISON

Baselines and Metrics. We compare with Random and state-of-the-art visual instruction selection methods under the same budget (Table 1). Detailed baseline descriptions are provided in Appendix C.1. We exclude methods that require changing the fixed `LLaVA-665K` candidate pool or introducing an additional data-construction or enhancement stage. Because benchmark scales differ, we report average relative performance (Rel.), computed relative to full-data fine-tuning. We also report total FLOPs (EFLOPs) and wall-clock time to assess efficiency.

Results. DATAVALVE achieves the best average relative performance, reaching 98.6% of full-data fine-tuning under a 20% target training budget. It obtains the best results on SQA-I, TextVQA, POPE, and LLaVA-Bench, and remains competitive on the others. These results show that DATAVALVE preserves broad multimodal capabilities rather than overfitting to a single benchmark. It requires 2.3 EFLOPs and 6.5h of wall-clock time. The 6.5h covers golden-set construction, CLIP extraction, and the complete integrated selection-and-training loop; see Appendix C.3.

4.3 ABLATION & PARAMETER STUDY

We validate the utility-gated reward, warm-up ratio, and golden set size. Appendix D provides additional studies of the window length H and target budget $\mathcal{B}$.

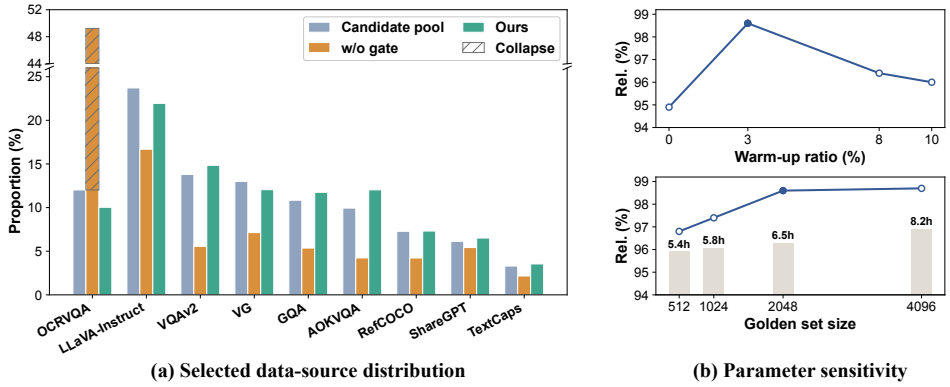
Utility Gate. We first ablate the online LoRA influence gate by replacing the full reward $R_t = \hat{\Delta}_t \gamma_t$ with $R_t = \hat{\Delta}_t$. This variant keeps the same golden set loss feedback but removes update-strength modulation, so the router is rewarded only by short-term golden-shard loss reduction.

Table 2 reports 98.6% Rel. for the full method and 95.0% without the utility gate. The no-gate variant remains competitive on TextVQA but degrades on broader multimodal benchmarks, indicating that loss-only feedback over-emphasizes a narrow high-loss source. As shown in Figure 3(a), OCRVQA rises from 12.0% of the candidate pool to 49.2% of routed samples without the gate. OCR-heavy samples often have larger losses and more immediate reduction space, so a loss-only reward can repeatedly reinforce OCRVQA to obtain fast golden loss decrease. The utility gate mitigates this collapse by down-weighting repeated OCR-heavy batches once their LoRA update strength falls below the recent local norm, reducing OCRVQA to 10.0% without an explicit diversity constraint.

Warm-up Ratio and Golden Set Size. Figure 3(b) shows that a small warm-up is important for stable online routing. Without warm-up, the router receives noisy early feedback before meaningful policy signals are available. A 3% warm-up performs best, while larger ratios spend too much of the

Table 2: **Ablation on the online LoRA influence gate.** We compare the full utility-gated reward with a no-gate variant that uses only the stabilized golden set loss feedback as the router reward.

Router Reward	VQAv2	GQA	SQA-I	TextVQA	POPE	MME	MMBench	MMBench(cn)	LLaVA-B	Rel. (%)
w/o gate: $R_t = \hat{\Delta}_t$	74.6	57.6	69.2	55.6	84.5	1407.9	60.7	53.1	57.6	95.0
Ours: $R_t = \hat{\Delta}_t \gamma_t$	77.2	59.7	70.8	56.7	87.5	1454.0	63.8	56.9	60.2	98.6



(a) Selected data-source distribution **(b) Parameter sensitivity**

Figure 3: **Ablation and sensitivity analysis. (a)** The no-gate variant collapses toward OCRVQA, while the full reward mitigates over-selection; hatching marks large OCRVQA excess over the candidate-pool proportion. **(b)** Sensitivity to warm-up ratio and golden set, supporting the default choice of $p_w = 3\%$ and $|\mathcal{G}| = 2048$.

fixed 20% budget on random data and leave less time for the router to learn. For golden set size, a very small set gives noisier shard feedback, while increasing $|\mathcal{G}|$ improves stability. The gain saturates around 2048 samples: 2048 and 4096 yield 98.6% and 98.7% Rel., respectively, while wall-clock time rises from 6.5h to 8.2h. Our default is $p_w = 3\%$ and $|\mathcal{G}| = 2048$.

Table 3: **Golden-set sampling strategy and source composition.**

Strategy	OCRVQA share (%)	VQAv2	GQA	SQA-I	TextVQA	POPE	MME	MMBench	MMBench(cn)	LLaVA-B	Rel. (%)
$[0, 100]$	12.02	76.5	59.2	69.5	55.8	86.2	1464.0	62.8	54.8	59.4	97.3
$[40, 90]$ (default)	15.72	77.2	59.7	70.8	56.7	87.5	1454.0	63.8	56.9	60.2	**98.6**
$[60, 90]$	20.75	76.2	58.5	69.4	55.2	86.0	1457.0	62.0	54.2	59.0	96.6

4.4 Further Analysis

Golden-Set Composition. Since DataValve relies on golden-set feedback to guide routing, we evaluate how different golden-set constructions affect downstream performance. We keep the golden set size fixed and compare sampling from three initial-loss percentile bands: $[0, 100]$, $[40, 90]$, and $[60, 90]$. Table 3 shows that the default $[40, 90]$ construction performs best, while both uniform sampling and the narrower $[60, 90]$ band reduce performance. Further analysis of the source mixtures reveals that the initial-loss distribution in `LLaVA-665K` is strongly source-skewed. For example, OCRVQA accounts for 12.02%, 15.72%, and 20.75% of the golden sets under the three strategies, respectively. The $[60, 90]$ band concentrates feedback on high-loss sources, whereas $[40, 90]$ retains broader source coverage while excluding trivial examples and extreme-loss outliers. Changing the loss-percentile band thus indirectly changes the golden set's source composition, not only its difficulty, with corresponding changes in downstream performance. The golden set therefore shapes which capabilities provide feedback to the router, linking holdout composition to the training data it learns to prioritize.

Cross-Dataset Transfer. We validate cross-dataset transfer on `Vision-Flan-186K` Xu et al. (2024), a diverse 187-task visual instruction pool with human-labeled examples and expert-written instructions (with a 20% budget). Table 4 shows that DataValve achieves 104.4% Rel., exceeding

Table 4: Evaluation on `Vision-Flan-186K` with LLaVA-v1.5-7B-LoRA.

Method	VQAv2	GQA	SQA-I	TextVQA	POPE	MME	MMB	MMB(cn)	LLaVA-B	Rel. (%)	Total FLOPs (EFLOPs)	Wall-clock Time (h)
Full-Finetune	74.5	47.1	61.8	46.4	85.7	1480.6	40.2	46.2	38.2	100.0	2.5	5.6
ICONS	75.5	45.1	62.4	45.3	83.0	1465.0	52.3	46.0	39.4	102.7	3.5	7.0
COIDO	75.4	44.7	63.1	45.6	83.5	1472.0	52.6	46.1	39.7	103.1	1.5	2.9
VISNEC	75.7	45.1	63.4	45.6	84.0	1458.0	52.4	46.7	**39.9**	103.4	2.0	2.3
Ours	**76.0**	**45.8**	**63.8**	**46.3**	**84.5**	**1485.0**	**53.9**	**47.0**	39.2	**104.4**	**0.6**	**2.1**

Table 5: Evaluation on `LLaVA-665K` with LLaVA-v1.5-13B-LoRA and Qwen3-VL-4B-Instruct.

Method	VQAv2	GQA	SQA-I	TextVQA	POPE	MME	MMB	MMB(cn)	LLaVA-B	Rel. (%)	Total FLOPs (EFLOPs)	Wall-clock Time (h)
LLaVA-v1.5-13B-LoRA												
Full	**80.0**	**63.3**	71.2	**60.2**	86.7	1541.7	**68.5**	**61.5**	60.7	100.0	18.9	42.2
Random	76.7	60.5	68.8	57.7	84.8	1484.9	62.8	55.2	57.4	94.9	**3.8**	**8.4**
Ours	77.2	60.3	**75.5**	59.5	**87.4**	**1573.5**	64.5	56.6	**63.4**	**98.9**	4.3	15.2
Qwen3-VL-4B-Instruct												
Full	84.0	63.3	72.3	83.3	**89.6**	2298.1	86.2	61.6	**73.4**	100.0	3.2	5.8
Random	83.6	63.0	67.4	83.8	89.0	2277.2	85.1	62.3	67.5	98.1	**0.6**	**1.2**
Ours	**85.0**	**65.6**	**75.6**	**83.9**	89.5	**2338.3**	**86.5**	**67.8**	**73.4**	**102.5**	0.7	2.1

VISNEC by 1.0 point, with the lowest FLOPs and shortest wall-clock time. All four selection methods exceed the full-data reference. Redundant and misaligned samples in `Vision-Flan-186K` can hinder learning. Selecting useful subsets reduces exposure to counterproductive updates, helping explain why less training data can yield better performance.

Cross-Backbone Transfer. We evaluate DATAVALVE on LLaVA-v1.5-13B-LoRA and Qwen3-VL-4B-Instruct Bai et al. (2025) using `LLaVA-665K`. The Qwen experiment continues LoRA tuning from its instruction-tuned checkpoint. Each setting learns its own router, with Random and DATAVALVE sharing a 20% budget and the same initialization, optimization, and evaluation protocol. Table 5 shows that DATAVALVE outperforms Random on both backbones, reaching 98.9% and 102.46% Rel., respectively, while using less computation and time than full-data tuning. At the same data budget, the gains over Random are 4.0 and 4.34 relative performance points, respectively, demonstrating the benefit of adaptive selection on both backbones.

Runtime Breakdown. The complete pipeline takes 6.5h, including 3min for golden-set construction, 1.5h for golden-shard feedback, and approximately 18min for CLIP extraction. Golden-shard feedback accounts for 23.1% of total runtime, making it the principal auxiliary target-model computation. Including all overhead, this is a 71.4% reduction from the 22.7h full-data run. Router scoring and updates are included in the integrated selection-and-training loop. With the golden-set size fixed, each feedback evaluation processes the same number of examples, whereas the full-pool target-model scoring pass avoided by DATAVALVE grows with candidate-pool size. Larger pools therefore offer greater potential absolute savings in computation and wall-clock time from eliminating this pass. Detailed component accounting is provided in Table 7 of Appendix C.3.

5 CONCLUSION

We present DATAVALVE, a selecting-while-training framework that integrates adaptive data selection into target MLLM fine-tuning. By coupling lightweight routing with golden-set feedback and relative LoRA update strength, DATAVALVE directs training computation toward currently useful samples without a separate full-pool target-model scoring stage. Experiments demonstrate strong performance and efficiency across candidate pools, model scales, and MLLM families. The gate ablation shows that update-strength modulation mitigates high-loss-source domination, while the golden-set construction study highlights the importance of feedback-set composition. Together, these findings support treating data selection as an adaptive training-time decision rather than fixed preprocessing. Our current implementation focuses on LoRA-based instruction tuning; future work will extend the framework to full-parameter fine-tuning and other post-training settings.

AI USE STATEMENT

Generative AI tools assisted with code development and language polishing. The authors take responsibility for the final content of this work.

ETHICS STATEMENT

This work studies data selection using existing multimodal datasets and models. Efficiency improvements do not remove biases or potentially harmful content in the underlying data, and golden-set composition can influence which capabilities are prioritized. Applications should respect dataset and model licenses and include appropriate safety and bias assessments. Further discussion appears in Appendix F.

REPRODUCIBILITY STATEMENT

Appendix A provides theoretical assumptions and proofs. Appendices B and C describe implementation details, hyperparameters, baselines, evaluation protocols, and efficiency accounting. Core code is included in the supplementary material; its README specifies the release scope and resources not included in this package.

REFERENCES

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48, 2009.

Jinhe Bi, Yifan Wang, Danqi Yan, Xun Xiao, Artur Hecker, Volker Tresp, and Yunpu Ma. PRISM: Self-pruning intrinsic selection method for training-free multimodal data selection. *arXiv preprint arXiv:2502.12119*, 2025.

Ruibo Chen, Yihan Wu, Lichang Chen, Guodong Liu, Qi He, Tianyi Xiong, Chenxi Liu, Junfeng Guo, and Heng Huang. Your vision-language model itself is a strong filter: Towards high-quality instruction tuning with data selection. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4156–4172. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.246.

Jiajun Deng, Zhengyuan Yang, Tianlang Chen, Wengang Zhou, and Houqiang Li. Transvg: End-to-end visual grounding with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1769–1779, 2021.

Mingkang Dong, Hongyi Cai, Jie Li, Sifan Zhou, Bin Ren, Kunyu Peng, and Yuqian Fu. Visnec: Measuring and leveraging visual necessity for multimodal instruction tuning. In *European Conference on Computer Vision*, 2026.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, 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.

Gemini Team, Google. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. URL https://arxiv.org/abs/2403.05530.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

Dongze Hao, Qunbo Wang, Longteng Guo, Jie Jiang, and Jing Liu. Self-bootstrapped visual-language model for knowledge selection and question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1857–1868, 2024.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1885–1894. PMLR, 2017.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pp. 1189–1197, 2010.

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, 2024. doi: 10.18653/v1/2024.emnlp-main.291.

Minjae Lee, Minhyuk Seo, Tingyu Qu, Tinne Tuytelaars, and Jonghyun Choi. Oasis: Online sample selection for continual visual instruction tuning. *arXiv preprint arXiv:2506.02011*, 2025.

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 Conference on Empirical Methods in Natural Language Processing*, 2023.

Xiaotian Lin, Yanlin Qi, Yizhang Zhu, Themis Palpanas, Chengliang Chai, Nan Tang, and Yuyu Luo. Lead: Iterative data selection for efficient llm instruction tuning. *Proceedings of the VLDB Endowment*, 19(3): 426–439, 2025. doi: 10.14778/3778092.3778103.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024a.

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? In *Computer Vision – ECCV 2024*, volume 15064, pp. 216–233, 2024b. doi: 10.1007/978-3-031-72658-3_13.

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. doi: 10.1145/3746027.3755160.

Pan Lu et al. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Pan Lu et al. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Adyasha Maharana, Jaehong Yoon, Tianlong Chen, and Mohit Bansal. Adapt-∞: Scalable continual multimodal instruction tuning via dynamic data selection. In *International Conference on Learning Representations*, 2025.

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.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14247–14256, 2025. doi: 10.1109/CVPR52734.2025.01329.

Burr Settles. Active learning literature survey. Technical Report 1648, University of Wisconsin–Madison, 2009.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 761–769, 2016.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, 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.

Jielin Song, Siyu Liu, Bin Zhu, and Yanghui Rao. Iterselecttune: An iterative training framework for efficient instruction-tuning data selection. *arXiv preprint arXiv:2410.13464*, 2024.

Jiachen T. Wang, Tong Wu, Dawn Song, Prateek Mittal, and Ruoxi Jia. GREATS: Online selection of high-quality data for LLM training in every iteration. In *Advances in Neural Information Processing Systems*, 2024a.

Wenxuan Wang, Yisi Zhang, Xingjian He, Yichen Yan, Zijia Zhao, Xinlong Wang, and Jing Liu. Beyond literal descriptions: Understanding and locating open-world objects aligned with human intentions. In *ACL (Findings)*, 2024b.

Lai Wei, Zihao Jiang, Weiran Huang, and Lichao Sun. Instructiongpt-4: A 200-instruction paradigm for fine-tuning minigpt-4. *arXiv preprint arXiv:2308.12067*, 2023.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992.

Changti Wu, Jiahuai Mao, Yuzhuo Miao, Shijie Lian, Bin Yu, Xiaopeng Lin, Cong Huang, Lei Zhang, and Kai Chen. Scalselect: Scalable training-free multimodal data selection for efficient visual instruction tuning. *arXiv preprint arXiv:2602.11636*, 2026.

Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163:21–40, 2017.

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.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54104–54132. PMLR, 2024a.

Tingyu Xia, Bowen Yu, Kai Dang, An Yang, Yuan Wu, Yuan Tian, Yi Chang, and Junyang Lin. Rethinking data selection at scale: Random selection is almost all you need. *arXiv preprint arXiv:2410.09335*, 2024b. URL https://arxiv.org/abs/2410.09335.

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. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.905.

Yichen Yan, Xingjian He, Sihan Chen, and Jing Liu. Calibration & reconstruction: Deeply integrated language for referring image segmentation. In *Proceedings of the 2024 International Conference on Multimedia Retrieval*, pp. 451–459, 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*, 2025.

Xiang Yue et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Ge Zheng, Bin Yang, Jiajin Tang, Hong-Yu Zhou, and Sibei Yang. DDCoT: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

A Theoretical Details

A.1 Formal Justification of the Update-Strength Proxy

Proposition 1 (Update-strength proxy). *Let $g_t = \nabla_{\theta^L} \ell_{\text{train}}(\theta_t; B_t)$ be the LoRA gradient induced by the routed batch B_t. We view the golden-shard loss as a function of the trainable LoRA parameters while holding the frozen backbone parameters fixed. Assume the golden-shard loss is differentiable around θ_t^L and the local LoRA update is $\theta_{t+1}^L = \theta_t^L - \eta_t^L g_t$, with $\eta_t^L \geq 0$. Define the ideal utility*

$$U_t = \ell_{\mathcal{G}_r}(\theta_t) - \ell_{\mathcal{G}_r}(\theta_{t+1}).$$

Then, under a first-order Taylor expansion,

$$U_t = \eta_t^L \nabla_{\theta^L} \ell_{\mathcal{G}_r}(\theta_t)^\top g_t + O\big((\eta_t^L)^2 \|g_t\|_2^2\big).$$

Let $U_t^{(1)}$ denote the leading inner-product term above. By Cauchy-Schwarz, the first-order magnitude satisfies

$$|U_t^{(1)}| \leq \eta_t^L \|\nabla_{\theta^L} \ell_{\mathcal{G}_r}(\theta_t)\|_2 \|g_t\|_2.$$

Thus, for a fixed feedback state and active golden shard, the routed-batch LoRA-gradient norm forms the action-dependent factor in an upper bound on the potential magnitude of the first-order utility.

Proof. By Taylor expansion of $\ell_{\mathcal{G}_r}$ at θ_t^L,

$$\ell_{\mathcal{G}_r}(\theta_{t+1}^L) = \ell_{\mathcal{G}_r}(\theta_t^L) + \nabla_{\theta^L} \ell_{\mathcal{G}_r}(\theta_t)^\top (\theta_{t+1}^L - \theta_t^L) + O\big(\|\theta_{t+1}^L - \theta_t^L\|_2^2\big).$$

Substituting $\theta_{t+1}^L - \theta_t^L = -\eta_t^L g_t$ gives

$$\ell_{\mathcal{G}_r}(\theta_{t+1}^L) - \ell_{\mathcal{G}_r}(\theta_t^L) = -\eta_t^L \nabla_{\theta^L} \ell_{\mathcal{G}_r}(\theta_t)^\top g_t + O\big((\eta_t^L)^2 \|g_t\|_2^2\big).$$

Rearranging yields the first statement. The inequality follows from Cauchy-Schwarz:

$$\big|\nabla_{\theta^L} \ell_{\mathcal{G}_r}(\theta_t)^\top g_t\big| \leq \|\nabla_{\theta^L} \ell_{\mathcal{G}_r}(\theta_t)\|_2 \, \|g_t\|_2.$$

□

For clarity, the result is stated under the local SGD-style update used in the main derivation. With adaptive optimizers such as AdamW, the argument applies analogously to the effective LoRA update direction, while g_t is used as a low-cost proxy for the parameter change norm. Importantly, this proposition does not assert that a larger LoRA-gradient norm makes the update beneficial. It only bounds the potential magnitude of the utility change. The sign is supplied by the observed golden set loss feedback $\hat{\Delta}_t$, while the positive gate γ_t only modulates credit strength.

Proposition 2 (Sign preservation and bounded credit modulation). *Let $R_t = \hat{\Delta}_t \, \gamma_t$ and $\gamma_t = 0.5 + \text{sigmoid}(z_t)$. Then $\gamma_t \in (0.5,\, 1.5)$. Consequently, if $\hat{\Delta}_t \neq 0$,*

$$\text{sign}(R_t) = \text{sign}(\hat{\Delta}_t),$$

and

$$0.5\,|\hat{\Delta}_t| < |R_t| < 1.5\,|\hat{\Delta}_t|.$$

Proof. Since $\text{sigmoid}(z_t) \in (0, 1)$, we have $\gamma_t = 0.5 + \text{sigmoid}(z_t) \in (0.5,\, 1.5)$. Hence $\gamma_t > 0$, so multiplying $\hat{\Delta}_t$ by γ_t cannot change its sign. The magnitude bound follows directly from $0.5 < \gamma_t < 1.5$. □

Explicit form of the short-window normalization. The $\text{Norm}_H(\cdot)$ operator used in the main text is defined as

$$\text{Norm}_H(s_t) = \frac{s_t - \mu_{t-1}}{\sqrt{v_{t-1}} + \epsilon},$$

where μ_{t-1} and v_{t-1} are the mean and variance of the H most recent s values before the current feedback event, and ϵ is a small numerical floor. We activate window-based normalization once the minimum history count is reached; before that, we set $\gamma_t = 1$. In implementation, we clamp the result to $[-2, 2]$ for numerical stability.

A.2 Advantage Baseline

The centered learning signal in Eq. 10 is computed from the previous H feedback events:

$$b_t = \frac{1}{H}\sum_{j=1}^{H} R_{t-j}, \qquad A_t = R_t - b_t.$$

During the first few feedback events, before H previous rewards are available, we use the available previous rewards once the minimum history count is reached; before that, we set $A_t = R_t$. Since b_t depends only on feedback values that precede the current routed action, it is independent of that action and serves as a standard policy-gradient baseline. We use the same window length H for the reward baseline and for the short-window normalization of LoRA update strength.

A.3 Policy-Gradient Surrogate for Top-k Routing

The router must choose exactly $k = |B_t|$ samples from each super-batch S_t. A fully stochastic independent Bernoulli policy, $\prod_i q_i^{m_i}(1-q_i)^{1-m_i}$, is therefore not the right action model: it produces a random number of selected samples and does not by itself satisfy the fixed training-budget constraint. One could instead condition the Bernoulli model on selecting exactly k samples,

$$\pi_\phi^{(k)}(B_t|S_t) = \frac{\prod_{z_i\in B_t} q_i \prod_{z_j\notin B_t}(1-q_j)}{Z_k(q)}, \qquad Z_k(q) = \sum_{\substack{C\subseteq S_t\\ |C|=k}} \prod_{z_i\in C} q_i \prod_{z_j\notin C}(1-q_j).$$

However, this exact fixed-cardinality likelihood is unattractive for online routing. Its normalization couples all candidates in S_t, is expensive to differentiate for every router update, and still treats all unselected samples as negative decisions. In our setting, unselected samples are not observed to be harmful; they are simply outside the deterministic Top-k budget for the current super-batch.

Using the unconditioned Bernoulli log-likelihood as a proxy would also be misaligned:

$$\log \pi_\phi(\mathcal{A}_t|S_t) = \sum_{z_i\in B_t} \log q_i + \sum_{z_j\notin B_t} \log(1-q_j).$$

Since $|S_t \setminus B_t| \gg |B_t|$, the unselected term can dominate the gradient. For positive advantage it pushes many logits downward merely because they were outside the budget, while for negative advantage it can push unselected logits upward. Both effects are undesirable for a rank-based Top-k router, whose goal is to increase or decrease credit for the selected action rather than to label every unselected candidate as bad.

We therefore use a policy-gradient-style surrogate over the selected action:

$$\mathcal{L}_{\mathrm{PG}} = -A_t \cdot \frac{1}{|B_t|}\sum_{z_i\in B_t} \log q_i.$$

Its gradient has the intended credit-assignment direction:

$$\nabla_\phi \mathcal{L}_{\mathrm{PG}} = -A_t \frac{1}{|B_t|}\sum_{z_i\in B_t} \nabla_\phi \log q_i.$$

Thus, when $A_t > 0$, minimizing the loss increases the probabilities of the selected samples; when $A_t < 0$, it decreases them. The fixed Top-k operator continues to enforce the training budget, while entropy and anchor regularization stabilize the global logit scale. We call the objective policy-gradient-style because it uses an advantage-weighted log-probability update for the selected action, but we do not claim it is the exact gradient of a fully stochastic Bernoulli policy. The effectiveness of this surrogate is supported by the empirical comparison of the full utility-gated policy to the no-gate variant in Section 4.3. This surrogate should be viewed as a biased but stable credit-assignment objective for deterministic Top-k routing, rather than an unbiased estimator of the gradient of the induced discrete selection rule.

B Implementation Details

B.1 Router Loss Regularization

The full router training objective is

$$\mathcal{L}_{\text{router}} = \mathcal{L}_{\text{PG}} - \lambda_H \mathcal{H}(q) + \lambda_A \left(\frac{1}{|S_t|} \sum_{z_i \in S_t} a_i - \text{logit}(\rho_t) \right)^2, \tag{13}$$

with $\lambda_H = \lambda_A = 0.02$, where $\mathcal{H}(q)$ is the Bernoulli entropy, a_i is the router logit before the sigmoid, and $\rho_t = |B_t|/|S_t|$. This objective keeps the policy-gradient signal as the main learning signal while adding two lightweight stabilizers. The entropy term prevents the router from becoming overconfident before it has received enough golden-shard feedback. The anchor term keeps the average router logit close to the logit of the target routing ratio, so the router learns a ranking preference without drifting toward globally selecting too many or too few candidates. We use a cosine router learning-rate schedule with a nonzero floor to keep late-stage policy updates active.

B.2 LoRA Gradient Capture

The online influence gate uses LoRA gradients that are already produced during ordinary target MLLM training. In practice, DeepSpeed ZeRO-style training may clear or shard parameter gradients before they can be inspected after the optimizer step. We therefore capture LoRA gradients during backward by registering hooks on LoRA parameters. The hooks detach and accumulate the current LoRA gradients for the feedback event while returning the original gradient unchanged, so target MLLM optimization is not modified. Router updates are likewise isolated from target MLLM optimization: delayed router scores are recomputed on stored features, and the policy-gradient loss is applied only to router parameters. For the first feedback event where g_{t-1} is unavailable, we set $\bar{g}_t = g_t$. Afterwards, the gate uses the two-step average $\bar{g}_t = (g_t + g_{t-1})/2$ and computes $\psi_t = \|\bar{g}_t\|_2^2$ as the update-strength statistic. The log-compressed update strength is appended to the recent-history window only after the current gate is computed, so the current batch does not enter its own baseline.

B.3 Shard-Aware Baseline for Golden Loss Feedback

The golden feedback $\hat{\Delta}_t$ normalizes the raw loss delta Δ_t using a shard-specific baseline rather than a single global statistic. The target MLLM may show different loss behavior on different shards at the same training stage, so a shared baseline would mix signals from qualitatively different data. Each shard independently tracks a small rolling window of its own recent losses, which provides a local reference for whether the post-update model state is improving or degrading on that shard.

B.4 Default Hyperparameters

Unless otherwise specified, we use a 20% target training budget, a 3% random warm-up ratio, a golden set of $|\mathcal{G}| = 2048$, and four rotating golden shards. The active shard is evaluated every 4 routed training steps in forward-only mode. Its loss baseline uses the previous four observations of that shard; stabilization divides by the recent shard-local loss scale with a numerical floor and symmetrically clips at three standard deviations. We set the gate and advantage history length to $H = 16$ with a minimum count of 4. When fewer than H previous values are available, we use all available values once the minimum count is reached; before that, we set $\gamma_t = 1$ and $A_t = R_t$. The target LoRA learning rate is 2×10^{-4} and the MM projector learning rate is 2×10^{-5}, following the standard LLaVA-v1.5 LoRA recipe. The router uses a four-layer MLP with hidden dimension 256, AdamW with learning rate 1.5×10^{-4}, a cosine schedule with minimum learning rate 5×10^{-5}, and optimizer accumulation over 4 feedback events. We set the router entropy and anchor weights to $\lambda_H = \lambda_A = 0.02$.

All main experiments are run on an Ubuntu server with eight NVIDIA RTX 5880 Ada Generation GPUs, each with 48GB memory. We use a per-GPU target MLLM batch size of 16. The router receives pre-extracted CLIP (`clip-vit-large-patch14-336`) features and outputs selected sample IDs for fetching the corresponding raw training examples.

Algorithm 1 DATAVALVE: Selecting-while-Training via Utility-Gated Routing

Input: Candidate pool $\mathcal{C}$, golden set $\mathcal{G}$, budget p, warm-up p_w, history H, feedback interval F
Split $\mathcal{G}$ into rotating shards $\{\mathcal{G}_r\}_{r=1}^{4}$
Warm up the target MLLM on random batches from $\mathcal{C}$; obtain θ_0
Initialize pending action $\mathcal{M} \leftarrow \emptyset$, histories, and step $s = 0$
while the routing budget is not exhausted **do**
  Sample $S_s \subseteq \mathcal{C}$, compute router scores, and select batch B_s
  Backpropagate target loss on B_s; capture LoRA gradient g_s
  if feedback slot (every F routed steps) **then**
    Compute ψ_s from g_s and the previous feedback-slot gradient
    Evaluate active shard at θ_s, before the current optimizer step: $L_r = \ell_{\mathcal{G}_r}(\theta_s)$
    if $\mathcal{M} \neq \emptyset$ **then**
      Retrieve prior action $(\tau, E_\tau, B_\tau, \psi_\tau)$ with $\tau < s$
      Assign delayed feedback $\hat{\Delta}_\tau = \mathrm{stabilize}(\mu_r - L_r)$
      Set $R_\tau = \hat{\Delta}_\tau \mathrm{RelGate}(\psi_\tau)$ and center to obtain A_τ
      Recompute scores on E_τ and update ϕ using Eq. 12
    end if
    Store current action $\mathcal{M} \leftarrow (s, E_s, B_s, \psi_s)$
    Update feedback histories and rotate the active shard
  end if
  Apply target optimizer step: $\theta_s \rightarrow \theta_{s+1}$; increment s
end while
Output: Trained target MLLM θ_s and router ϕ

C EXPERIMENTAL DETAILS

C.1 BASELINES

We compare DATAVALVE with simple heuristics and prior visual-instruction selection methods. For fair comparison, all included methods operate on the same fixed `LLaVA-665K` visual-instruction candidate pool. We exclude methods that require modifying the candidate pool or adding a separate data construction or enhancement stage, since those settings change the selection protocol.

Random selection samples the target-budget subset uniformly from the candidate pool.

SELF-FILTER Chen et al. (2024) uses target model feedback to filter low-quality instruction data. COINCIDE Lee et al. (2024) selects representative multimodal examples using concept and relevance signals. TIVE Liu et al. (2025) estimates instance utility through gradient-based influence signals. ICONS Wu et al. (2024) aggregates influence consensus across tasks to identify broadly useful samples. COIDO Yan et al. (2025) trains a lightweight scorer from a subset and combines importance with diversity when selecting the final training data. VISNEC Dong et al. (2026) scores visual necessity from the difference between blind and multimodal target-model losses and applies semantic-cluster stratification.

C.2 EVALUATION PROTOCOL

We evaluate downstream performance on the same benchmark suite used in the main paper: VQAv2, GQA, SQA-I, TextVQA, POPE, MME, MMBench-en, MMBench-cn, and LLaVA-Bench. Since metrics have different scales, we report average relative performance, computed by normalizing each method to the corresponding Full-Finetune result on each benchmark and averaging across benchmarks. LLaVA-Bench is evaluated with the same fixed LLM judge snapshot for all methods.

C.3 COST ANALYSIS DETAILS AND RUNTIME PROFILE

Table 7 reports the 6.5h end-to-end wall-clock time and selected component costs under the default 20% budget. The total includes golden-set construction, CLIP feature extraction, and the complete integrated selection-and-training loop, including router scoring, router updates, and golden-shard

Table 6: **Descriptions of benchmarks used for MLLM evaluation.**

Benchmark	Description
VQAv2 (Goyal et al., 2017)	Evaluates open-ended visual question answering on natural images, requiring models to recognize objects, attributes, and visual relations.
GQA (Hudson & Manning, 2019)	Tests compositional visual reasoning with questions grounded in scene graphs, emphasizing relations, attributes, and multi-step inference.
SQA-I (Lu et al., 2022)	Measures science-oriented and situated reasoning over images, covering questions that require both visual evidence and external commonsense knowledge.
TextVQA (Singh et al., 2019)	Focuses on OCR-aware visual question answering, where models must read and reason over text appearing in the image.
POPE (Li et al., 2023)	Probes object hallucination by asking presence/absence questions, measuring whether the model can avoid over-asserting unseen objects.
MME (Fu et al., 2023)	Provides a broad perception-and-cognition evaluation suite for multimodal models, reported as an aggregate score over diverse visual tasks.
MMBench-en (Liu et al., 2024b)	Evaluates English multimodal understanding with multiple-choice questions covering perception, reasoning, and knowledge-intensive visual tasks.
MMBench-cn (Liu et al., 2024b)	Uses the Chinese split of MMBench to assess multilingual multimodal understanding under the same benchmark design.
LLaVA-Bench (Liu et al., 2023)	Evaluates free-form multimodal generation and instruction following with an LLM-as-a-judge protocol.

Table 7: **Total runtime and component accounting at the default 20% budget.**

Component	Time	Share of total	Accounting note
Total time	6.5h	100%	Preparation and the complete selection-and-training loop.
Golden-set split	3min	0.8%	Initial-loss evaluation and golden-set construction.
Golden-shard feedback	1.5h	23.1%	Forward-only evaluations during training.
CLIP extraction	$\sim$18min	$\sim$4.6%	Candidate-feature pre-extraction.

feedback. Golden-set construction takes 3 minutes, and CLIP extraction takes approximately 18 minutes; both are included in the reported total. The component rows are selected costs within the total, not a separate selection stage or an exhaustive partition of training time.

D Additional Results

D.1 Recent-History Window Length

Table 8 studies the recent-history window length H, which is shared by the relative LoRA gate and the reward baseline. Smaller windows react faster but can be noisier, while larger windows are smoother but may respond more slowly to training-stage changes. We use $H = 16$ as the default, which balances local responsiveness and feedback stability.

D.2 Target Budget $\mathcal{B}$

Table 9 studies the effect of the target budget $\mathcal{B}$. Performance improves with more data, with the strongest gain between 10% and 20%; beyond 20%, gains gradually saturate toward full-data performance.

Table 8: **Ablation on recent-history window length H.**

Window Length	VQAv2	GQA	SQA-I	TextVQA	POPE	MME	MMBench	MMBench(cn)	LLaVA-B	Rel. (%)
$H = 8$	76.1	58.5	69.6	55.7	86.1	1461.1	62.8	55.1	60.0	97.2
$H = 16$ (default)	77.2	59.7	70.8	56.7	87.5	1454.0	63.8	56.9	60.2	98.6
$H = 32$	75.9	58.2	69.4	55.4	86.0	1454.0	62.5	54.8	59.4	96.7

Table 9: **Ablation on target training budget.**

Budget	VQAv2	GQA	SQA-I	TextVQA	POPE	MME	MMBench	MMBench(cn)	LLaVA-B	Rel. (%)	FLOPs	Time
Full	79.1	63.0	68.4	58.2	86.4	1476.9	66.1	58.9	59.3	100.0	10.2	22.7h
10%	75.9	58.4	67.9	54.7	85.1	1454.1	60.2	52.8	57.7	95.2	1.1	3.4h
20%	77.2	59.7	70.8	56.7	87.5	1454.0	63.8	56.9	60.2	98.6	2.3	6.5h
30%	77.7	60.4	70.4	57.2	86.7	1463.0	64.1	56.5	60.2	98.8	3.5	9.6h
40%	77.5	61.3	70.7	57.4	86.4	1478.0	64.9	57.7	59.3	99.3	4.7	12.5h

E LIMITATIONS

DATAVALVE is designed for efficient visual instruction tuning under a fixed candidate pool and a target training budget. Its feedback signal relies on a small internal golden set, so the quality and coverage of this holdout can affect routing behavior. We reduce this sensitivity by using a small 2048-example golden set, rotating shards, short-window normalization, and a studies of golden set size and construction, but highly specialized candidate pools may still require checking that the golden set covers the intended capabilities.

Our experiments cover LLaVA and Qwen3-VL models under LoRA-based target training. Full-parameter tuning and broader validation on multilingual or domain-specific instruction pools remain future work. Finally, DATAVALVE reduces target MLLM scoring and training cost, but it does not eliminate the cost of online training: golden set evaluations, feature extraction, and router updates still add overhead relative to pure random subset training.

F BROADER IMPACT

DATAVALVE aims to make visual instruction tuning more compute-efficient by selecting useful samples while the target MLLM is being trained. This can lower the cost of adapting open multimodal models and may make empirical research more accessible to groups with limited compute. It can also reduce unnecessary target model training passes, which has positive implications for energy use when applied responsibly.

The same efficiency gains could also make it easier to tune multimodal models for undesirable applications, or to amplify biases already present in the candidate data. Since DATAVALVE optimizes selection from an existing instruction pool rather than auditing or sanitizing that pool, it should be used together with standard dataset governance, safety filtering, and downstream evaluation. The router's online feedback can also shift selection toward sources that are overrepresented in the golden feedback, so monitoring source distributions and task-level behavior is important when deploying the method beyond controlled research settings.