THE VALUE OF TRAINING DATA IS NOT STATIC: OPTIMIZATION-AWARE ADAPTIVE SELECTION FOR MULTIMODAL INSTRUCTION TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal instruction tuning relies on large-scale image-text datasets, making effective data selection critical for improving training efficiency. Existing approaches mostly adopt static offline scoring or measure visual dependence at inference time, yet these metrics cannot directly quantify how a training example's gradient update will benefit the current evolving multimodal model. To address this gap, we propose **DOVAS**, a region-based dynamic data-allocation framework that combines offline visual-evidence scoring with online optimizer-aligned utility estimation. DOVAS first partitions the instruction corpus into fixed, semantically stratified regions. During training, it periodically re-weights regional sampling probabilities by estimating the predicted impact of candidate optimizer updates without modifying region membership. This design lets the model draw adaptively useful training data throughout the optimization trajectory. Experiments on standard multimodal instruction-tuning benchmarks demonstrate that DOVAS consistently outperforms competitive static data-selection baselines. Using only half of the full-data training compute budget, it surpasses the performance of full-dataset training across multiple model families. These results demonstrate that the training-time value of multimodal samples is not a static intrinsic property but depends on model state and optimizer dynamics.

1 INTRODUCTION

Visual instruction tuning trains multimodal large language models (MLLMs) to follow instructions about images, enabling visual question answering, grounded conversation, and compositional reasoning (Liu et al., 2023; Dai et al., 2023; Bai et al., 2023; Zhu et al., 2024). Larger instruction datasets increase training cost, but do not always yield proportional gains (Liu et al., 2024; Zhao et al., 2023; Xu et al., 2024). They can contain repetitive tasks, questions answerable without the image, and responses weakly supported by visual evidence. Selecting useful examples from these collections is important for efficient training.

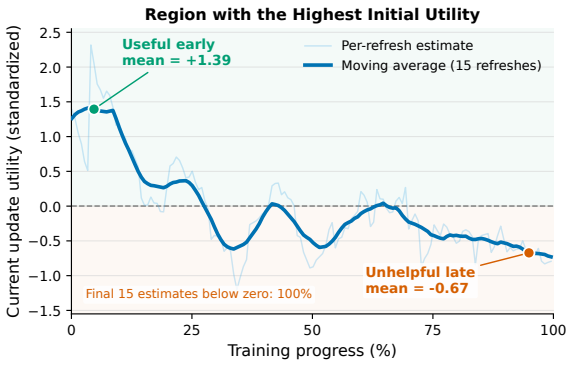


Figure 1: **A region's relative utility changes during training.** The region with the highest initial utility falls below the regional average later in training. Values are standardized across regions and smoothed; a negative value indicates lower relative utility, not necessarily a harmful update. Its membership remains fixed.

Static data selection methods use offline metrics to organize the instruction pool and select a fixed training subset. Selection criteria include quality and preference scores (Wei et al., 2023; Chen et al., 2024; Huang et al., 2025; Wu et al., 2026a; Dong et al., 2026a). Other methods consider diversity, task influence, or cross-modal alignment (Lee et al., 2024; Safaei et al., 2025; Yan et al., 2026; Wu et al.,

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.

2026b; Liu et al., 2025; Wu et al., 2024; Naharas et al., 2025). Recent approaches also measure visual dependence (Lee & Hwang, 2026; Dong et al., 2026b). However, restricting training to a small subset limits coverage, while repeated reuse can lead to diminishing returns or overfitting. More fundamentally, an offline assessment cannot track how the value of that subset changes during training. Figure 1 illustrates this issue: a region favored early in training becomes less valuable relative to others later. This observation motivates adapting the training distribution as the model learns.

Dynamic methods incorporate model-dependent losses, learning progress, or optimizer-aware utility estimates into data selection (Mindermann et al., 2022; Chen et al., 2023; Evans et al., 2024; Wang et al., 2026). These methods need to repeatedly rank candidate examples with the current model, creating a trade-off between frequent adaptation and selection cost (Evans et al., 2024; Wang et al., 2026). Moreover, the design of scoring objective is equally important. In visual instruction tuning under dynamic training data pools, a lower answer loss does not always reveal whether an update improves the use of image evidence or merely strengthens predictions available from the text. An adaptive selector should assess both answer improvement and the advantage of the correct image over missing or incorrect images, while keeping the cost of reassessment within the training budget.

To address these issues, we introduce **D**ynamic **O**ptimization-aligned **V**isual d**A**ta **S**election (DOVAS), which dynamically estimates the value of updates from fixed, semantically stratified evidence regions using proxy objectives under different visual specific conditions. Before training, we group instructions by semantics and partition them using answer compatibility, visual gain, and correct-image specificity scores. During training, we estimate how a candidate optimizer update from each region would change the objectives on samples. These estimates determine how often each region is sampled, while its membership stays fixed. The model therefore receives a fine-grained, state-dependent training mixture without repeatedly rebuilding the subset or changing the underlying region boundaries. Across the evaluated datasets and models, DOVAS improves aggregate performance over the compared selection baselines and matches or exceeds full-data performance.

Our contributions are as follows.

- We identify a central limitation of existing multimodal data selection. Scores computed before training describe data under a fixed model state, but do not determine whether the resulting updates remain useful to the evolving target model.
- We introduce DOVAS, a dynamic multimodal data selection framework that fixes semantically stratified evidence regions offline and reallocates their sampling probabilities using the current model and optimizer states. This separation combines stable data boundaries with fine-grained dynamic control of the training mixture.
- We evaluate DOVAS against full-data training and strong static selection methods across a broad set of multimodal tasks. The results demonstrate improved performance and training efficiency, and provide a new perspective on how data selection can interact with the learning dynamics of MLLMs.

2 Related Work

2.1 Visual Instruction Tuning

Multimodal models connect visual encoders to language models through learned interfaces. Flamingo inserts cross-attention layers into a frozen language model, while BLIP-2 maps visual features to a frozen language model through a Querying Transformer (Alayrac et al., 2022; Li et al., 2023). LLaVA trains a lightweight projection module and fine-tunes on visual instructions generated by a language model (Liu et al., 2023). MiniGPT-4 and InstructBLIP show how instruction tuning improves visual dialogue and generalization across vision-language tasks (Zhu et al., 2024; Dai et al., 2023). Subsequent work broadens supervision: SVIT adds complex reasoning (Zhao et al., 2023), Otter supports multi-turn interaction (Li et al., 2025), Vision-Flan collects human-labeled instructions across tasks (Xu et al., 2024), and LLaVA-1.5 combines public datasets (Liu et al., 2024). Qwen-VL extends capabilities to localization and text-rich images (Bai et al., 2023). Together, these efforts improve capability through the source, coverage, and composition of instruction data. Our work studies how to allocate an existing instruction pool as training progresses.

2.2 Data Selection

Data selection improves efficiency by prioritizing useful supervision. DSIR matches a target text distribution through importance resampling (Xie et al., 2023). For visual instruction tuning, InstructionGPT-4, Self-Filter, Align^{2}LLaVA, and DOSE assess quality, difficulty, preference, or image-text alignment (Wei et al., 2023; Chen et al., 2024; Huang et al., 2025; Wu et al., 2026a). Once-For-All learns selectors reusable across data pools and models (Dong et al., 2026a), while coverage-based methods preserve representative images, concept diversity, or representation subspaces (Lee et al., 2024; Safaei et al., 2025; Yan et al., 2026; Wu et al., 2026b). These approaches emphasize the properties and coverage of selected data.

Optimization-based methods assess learnability or relations among skills and examples, as in loss-based selection, Skill-It, and JEST (Mindermann et al., 2022; Chen et al., 2023; Evans et al., 2024). LESS measures optimizer-aware gradient similarity to target examples, and OPUS estimates update utility during pretraining (Xia et al., 2024; Wang et al., 2026). For visual instruction tuning, TIVE and ICONS estimate task influence, while XMAS uses a proxy model's cross-modal alignment trajectories (Liu et al., 2025; Wu et al., 2024; Naharas et al., 2025).

VIG and VisNec measure how much images reduce predictive uncertainty (Lee & Hwang, 2026; Dong et al., 2026b). Such visual-dependence scores do not directly measure how training updates affect other examples. Building on these signals and optimizer-aware selection, DOVAS fixes semantically stratified evidence regions offline and adjusts their sampling probabilities using estimated update effects on answer and visual counterfactual objectives. The partition remains unchanged while allocation adapts to the target model.

3 Method

Figure 2 summarizes DOVAS. Offline scoring organizes the data into fixed evidence regions within semantic clusters and constructs held-out proxy pools. During training, we use regional probes and the current optimizer state to estimate update utility on these proxies. The estimates guide sampling for the next training block, and this process repeats as the model learns.

3.1 Preliminaries

Visual instruction tuning uses a dataset $\mathcal{D} = \{(v_i, x_i, y_i)\}_{i=1}^{N}$, where v_i is an image, x_i is an instruction, and $y_i = (y_{i,1}, \ldots, y_{i,T_i})$ is the target response. A vision encoder and cross-modal connector provide visual tokens to an autoregressive language model (Liu et al., 2023; Dai et al., 2023; Liu et al., 2024). Teacher forcing supervises only response tokens, masking image and instruction labels:

$$\ell_{\mathrm{MM}}(\theta; v_i, x_i, y_i) = -\frac{1}{T_i} \sum_{t=1}^{T_i} \log p_\theta(y_{i,t} \mid v_i, x_i, y_{i,<t}). \tag{1}$$

For a mini-batch $\mathcal{B}$, the training objective is $\mathcal{L}_{\mathrm{MM}}(\theta; \mathcal{B}) = |\mathcal{B}|^{-1} \sum_{i \in \mathcal{B}} \ell_{\mathrm{MM}}(\theta; v_i, x_i, y_i)$. Starting from an aligned checkpoint, we train the connector and language model, denoted by θ, while freezing the visual encoder. For batch $\mathcal{B}_k$, $g_k = \nabla_\theta \mathcal{L}_{\mathrm{MM}}(\theta_k; \mathcal{B}_k)$ denotes the training gradient and s_k the optimizer state. The update is

$$(\theta_{k+1}, s_{k+1}) = \mathrm{Opt}(\theta_k, s_k, g_k). \tag{2}$$

DOVAS keeps the model, loss, and optimizer unchanged. It partitions the training pool into fixed regions and adjusts their sampling probabilities as training progresses.

3.2 Offline Multimodal Signals

DOVAS groups instructions into semantic clusters using a pretrained text encoder; c_i denotes sample i's cluster. Scores are compared within clusters to account for differences in instruction intent. Before training, a fixed multimodal scorer O evaluates each sample with its original image, without

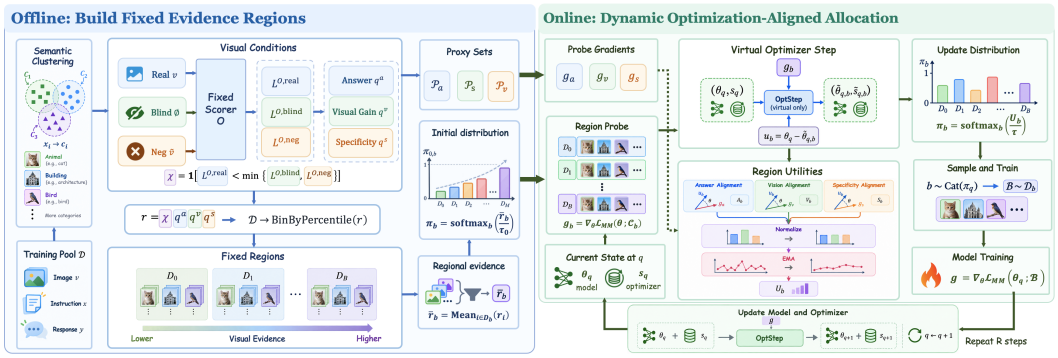


Figure 2: **DOVAS pipeline.** Offline preparation fixes the evidence regions and proxy pools. Online allocation estimates regional update utility using answer and visual counterfactual objectives, then standardizes and smooths these estimates to update sampling probabilities. Virtual updates are discarded; sampled training batches update the model.

an image, and with a semantically matched negative image:

$$\begin{aligned} L_i^{O,\text{real}} &= \ell_{\text{MM}}(O; v_i, x_i, y_i), \\ L_i^{O,\text{blind}} &= \ell_{\text{MM}}(O; \varnothing, x_i, y_i), \\ L_i^{O,\text{neg}} &= \ell_{\text{MM}}(O; \tilde{v}_i, x_i, y_i). \end{aligned} \tag{3}$$

The blind condition keeps the instruction and response unchanged. To select the negative image $\tilde{v}_i$, we search within c_i for similar instructions with different images and non-overlapping answers. A pair is retained only when swapping its images in both directions increases the corresponding response loss.

Based on these losses, the real-image loss measures answer compatibility, $\Delta_i^v = L_i^{O,\text{blind}} - L_i^{O,\text{real}}$ measures visual gain, and $\Delta_i^s = L_i^{O,\text{neg}} - L_i^{O,\text{real}}$ measures specificity to the correct image. Since their scales vary across instruction types, we convert the three measurements into percentile scores within each semantic cluster

$$q_i^a = 1 - \text{Percentile}_{c_i}(L_i^{O,\text{real}}), \qquad q_i^v = \text{Percentile}_{c_i}(\Delta_i^v), \qquad q_i^s = \text{Percentile}_{c_i}(\Delta_i^s). \tag{4}$$

Higher q_i^a, q_i^v, and q_i^s indicate stronger answer compatibility, visual gain, and correct-image specificity, respectively.

3.3 Static Partition of the Training Pool

We first require the correct image to give a lower response loss than either counterfactual input:

$$\chi_i = \mathbf{1}\left[L_i^{O,\text{real}} < \min\left\{L_i^{O,\text{blind}}, L_i^{O,\text{neg}}\right\}\right]. \tag{5}$$

The sample-level evidence score combines this condition with the three scores:

$$r_i = \chi_i q_i^a q_i^v q_i^s. \tag{6}$$

This product gives a high value only when the sample receives strong evidence from all three measurements. Samples that do not satisfy the real-image condition form a separate region within each semantic cluster. The remaining samples are divided into M equal-frequency regions according to r_i. Let $\text{Percentile}_{c_i,\chi=1}(r_i)$ denote the percentile of r_i among samples satisfying $\chi_i = 1$ in cluster c_i. The region index is

$$b_i = \begin{cases} 0, & \chi_i = 0, \\ 1 + \min\{M-1, \lfloor M\, \text{Percentile}_{c_i,\chi=1}(r_i) \rfloor\}, & \chi_i = 1, \end{cases} \qquad \mathcal{D}_b = \{i : b_i = b\}. \tag{7}$$

The boundaries remain fixed during training, giving each region a comparable evidence range across semantic clusters. Its mean evidence score is

$$\bar{r}_b = \frac{1}{|\mathcal{D}_b|} \sum_{i \in \mathcal{D}_b} r_i. \tag{8}$$

The first training block uses an initial allocation obtained with a softmax over the regional scores

$$\pi_{0,b} = \mathrm{softmax}_b(\bar{\mathbf{r}}/\tau_0)\,, \tag{9}$$

where τ_0 controls the concentration of the initial distribution. We keep the original update-token share ρ_c of each semantic cluster to preserve semantic coverage. We also reserve held-out proxy pools stratified by semantic cluster and evidence region. Samples with high q_i^a, q_i^v, and q_i^s form $\mathcal{P}_a$, $\mathcal{P}_v$, and $\mathcal{P}_s$, respectively, with $\mathcal{P}_s$ restricted to verified negative pairs. These samples are drawn only from the training pool and excluded from parameter training. The fixed regions and proxy pools remain fixed throughout training and define the common reference used to compare training states. At each allocation refresh, we draw mini-batches from these pools to estimate the proxy gradients.

3.4 Dynamic Optimization-Aligned Allocation

Dynamic visual objectives. DOVAS refreshes regional sampling probabilities every R optimizer steps after the initial block. At refresh q, let θ_q and s_q denote the model and optimizer states. Using the losses in Equation 3 with θ_q in place of the offline scorer O, we define the objectives over the fixed proxy pools as

$$J_a(\theta_q) = \frac{1}{|\mathcal{P}_a|} \sum_{i \in \mathcal{P}_a} L_i^{\theta_q,\mathrm{real}}, \tag{10}$$

$$J_v(\theta_q) = \frac{1}{|\mathcal{P}_v|} \sum_{i \in \mathcal{P}_v} \left(L_i^{\theta_q,\mathrm{real}} - L_i^{\theta_q,\mathrm{blind}} \right), \tag{11}$$

$$J_s(\theta_q) = \frac{1}{|\mathcal{P}_s|} \sum_{i \in \mathcal{P}_s} \left(L_i^{\theta_q,\mathrm{real}} - L_i^{\theta_q,\mathrm{neg}} \right). \tag{12}$$

The answer objective measures the response loss with the correct image. The visual objective measures the loss difference between the real and blind conditions, while the specificity objective measures the difference between the real and negative conditions. A smaller difference indicates a stronger preference for the correct image. These objectives are used only to obtain local gradient directions, so no additional margin or nonlinear transformation is required.

We estimate these gradients using a newly drawn proxy mini-batch $\mathcal{Q}_x^{(q)} \subset \mathcal{P}_x$ for each objective $x \in \{a, v, s\}$ at refresh q. Let $\ell_x(\theta; i)$ denote the corresponding per-example summand in the pool objectives above. The sampled objective and its stochastic gradient are

$$\widehat{J}_x^{(q)}(\theta) = \frac{1}{|\mathcal{Q}_x^{(q)}|} \sum_{i \in \mathcal{Q}_x^{(q)}} \ell_x(\theta; i), \qquad g_x^{(q)} = \nabla_\theta \widehat{J}_x^{(q)}(\theta_q). \tag{13}$$

The default mini-batch size is $|\mathcal{Q}_x^{(q)}| = 16$ per objective per refresh; this is not the size of $\mathcal{P}_x$. Each sampled gradient is shared across all regional utility evaluations at the same refresh.

Optimizer-aligned update utility. To estimate the effect of training on a region, we draw a probe batch $\mathcal{C}_b^{(q)} \subset \mathcal{D}_b$ with semantic stratification and compute its standard instruction-tuning gradient

$$g_b^{(q)} = \nabla_\theta \mathcal{L}_{\mathrm{MM}}(\theta_q; \mathcal{C}_b^{(q)}). \tag{14}$$

Raw gradient similarity does not capture how the optimizer transforms a candidate gradient. We therefore apply one virtual optimizer step to each region gradient using a copy of the current state s_q

$$(\theta_{q,b}^+, s_{q,b}^+) = \mathrm{Opt}(\theta_q, s_q, g_b^{(q)}), \qquad u_b^{(q)} = \theta_q - \theta_{q,b}^+. \tag{15}$$

The virtual step is discarded after measuring $u_b^{(q)}$ and does not change the model or optimizer. It uses the same optimizer state, learning rates, and update rule as training. We then define three regional update utilities

$$A_b^{(q)} = \langle u_b^{(q)}, g_a^{(q)} \rangle, \quad V_b^{(q)} = \langle u_b^{(q)}, g_v^{(q)} \rangle, \quad S_b^{(q)} = \langle u_b^{(q)}, g_s^{(q)} \rangle. \tag{16}$$

Since $\theta_{q,b}^+ = \theta_q - u_b^{(q)}$, first-order expansion gives $J(\theta_{q,b}^+) \approx J(\theta_q) - \langle u_b^{(q)}, \nabla J(\theta_q) \rangle$. Using $g_x^{(q)}$ in place of the full-pool gradient gives a stochastic estimate of this first-order improvement.

A positive utility predicts a reduction in the sampled proxy objective; agreement with the full pool depends on mini-batch estimation error. These objectives are allocation proxies and do not guarantee improvements in visual grounding.

Dynamic region allocation. The three utility families can have different scales. At each refresh, we standardize each family across regions using its mean $\mu_X^{(q)}$ and standard deviation $\sigma_X^{(q)}$

$$Z_{X,b}^{(q)} = \frac{X_b^{(q)} - \mu_X^{(q)}}{\sigma_X^{(q)}}, \qquad X \in \{A, V, S\}. \tag{17}$$

We smooth the standardized utilities across consecutive refreshes

$$\widehat{X}_b^{(q)} = \beta \widehat{X}_b^{(q-1)} + (1-\beta) Z_{X,b}^{(q)}, \qquad X \in \{A, V, S\}, \quad q > 1. \tag{18}$$

At the first refresh, $\widehat{X}_b^{(1)} = Z_{X,b}^{(1)}$. We combine the three utilities and update the region distribution as

$$U_b^{(q)} = \widehat{A}_b^{(q)} + \widehat{V}_b^{(q)} + \widehat{S}_b^{(q)}, \qquad \pi_b^{(q)} = \mathrm{softmax}_b\Big(U_b^{(q)}/\tau\Big). \tag{19}$$

The initial distribution π_0 is used only before the first refresh. After that, the temperature τ controls how strongly the measured utilities change the allocation. Thus, the offline partition supplies a fixed structural constraint, while $\pi^{(q)}$ provides fine-grained control of the training mixture as the model and optimizer evolve. An optimization interpretation of the allocation rule under fixed measured utilities is provided in Appendix A.

For the following R updates, we first sample a region according to $\pi^{(q)}$. Within the selected region, each batch follows the original semantic-cluster shares ρ_c, and samples are drawn uniformly from the corresponding region–cluster cell

$$i \sim \mathrm{Uniform}(\mathcal{D}_b \cap \mathcal{D}_c). \tag{20}$$

Here $\mathcal{D}_c = \{i : c_i = c\}$ denotes the samples in semantic cluster c. The region distribution controls which evidence levels receive more updates, while ρ_c preserves semantic coverage. Uniform sampling within each cell avoids introducing an additional static preference after the region has been selected. The model is updated with the standard loss in Equation 1.

4 Experiments

4.1 Experimental Setup

Datasets and Models. We conduct experiments on LLaVA-665K (Liu et al., 2024) and Vision-Flan-186K (Xu et al., 2024). LLaVA-665K contains approximately 665K multimodal instructions drawn from a compact mixture of visual question answering, reasoning, grounding, OCR, and conversation tasks. Vision-Flan-186K contains 186K examples from 191 tasks and provides a more diverse task distribution with fewer examples per task.

We use Qwen3.5-9B (Qwen Team, 2026) as the scoring model for computing the visual contrast metrics. We use LLaVA-v1.5-7B (Liu et al., 2024) as the primary model for comparison with existing selection methods. We further evaluate Qwen2.5-VL-7B and Qwen3.5-4B (Qwen Team, 2026) to examine whether the method remains effective across model families and scales.

Compute budget. We define the online compute budget to include parameter training and online selection, using the FLOP accounting in Appendix D. Unless otherwise stated, selection methods use a cap of $B = 0.5C_{\text{full}}$, where C_{full} is the training compute of the full-data baseline for the same dataset and target model. Offline annotation and preparation are excluded for all methods. Online-selection costs are charged against this cap, so methods may perform different numbers of parameter updates under the same online budget.

4.2 Main Results

Tables 1 and 2 compare DOVAS with full-data training and existing selection methods, including OPUS, on the two instruction-tuning datasets. DOVAS combines fixed evidence regions with

Table 1: Overall performance on LLaVA-665K using LLaVA-v1.5-7B with full-parameter fine-tuning. All selection methods use a matched online compute budget equal to 50% of that used for full-data training, while Full Data uses the entire training set. Offline annotation and preparation are excluded. Rel. is the mean of benchmark-wise performance normalized by Full Data. The best result in each column is shown in bold.

Method	VQAv2	GQA	LLaVA-W	SQA-I	TextVQA	MME-P	MMB-EN	MMB-CN	POPE	MM-Vet	Rel.
Full Data	78.5	62.0	65.4	66.8	58.2	1510.7	64.3	58.3	85.9	31.1	100.0%
Random	74.5	56.3	58.8	67.8	56.2	1423.2	62.3	54.1	85.4	31.7	95.9%
Self-Filter	74.6	55.9	63.1	64.2	52.9	1399.8	53.3	48.8	86.9	29.7	92.5%
EL2N	76.8	57.2	61.1	67.5	51.4	1429.3	61.1	51.1	83.2	31.1	94.7%
TypiClust	77.2	59.9	63.3	64.4	54.5	1422.3	63.5	56.9	85.2	31.3	97.2%
IFD	75.8	57.4	62.1	66.7	52.5	1355.3	58.5	51.8	88.0	28.2	93.7%
PreSel	76.3	59.1	62.6	**71.0**	54.9	1459.8	63.9	55.9	85.6	32.3	98.4%
XMAS	75.4	61.3	60.3	67.5	54.4	1479.4	63.8	53.2	87.4	**32.5**	97.6%
COINCIDE	76.4	58.4	65.1	69.0	54.7	1441.2	60.8	53.9	86.9	28.9	96.5%
CLIP-Score	74.5	56.9	61.8	66.8	54.8	1423.5	54.9	52.1	85.9	31.1	94.4%
CoIDO	75.9	61.1	64.9	67.2	57.7	1442.2	63.5	55.2	87.1	30.9	98.4%
ICONS	76.9	61.1	64.5	67.9	54.9	1446.2	62.9	55.8	**88.5**	30.2	98.1%
ScalSelect	76.9	61.7	62.9	66.1	58.1	1489.9	62.9	56.7	85.0	31.2	98.5%
DOSE	75.7	61.1	62.9	64.9	57.0	1479.6	61.9	57.2	83.9	31.0	97.6%
Once-For-All	77.1	60.9	64.4	65.7	56.9	1502.9	62.9	56.8	86.2	31.1	98.6%
TIVE	76.8	60.9	63.9	64.8	58.1	1482.5	62.9	56.8	82.9	31.1	98.0%
OPUS	75.6	61.2	64.1	64.6	59.0	1471.4	62.5	57.1	83.2	30.6	97.9%
VIG	77.8	61.2	65.2	66.0	58.3	1494.7	64.2	57.1	86.1	30.9	99.3%
VisNec	78.0	59.2	66.8	69.4	56.2	1460.9	64.2	**59.3**	86.9	32.2	100.0%
DOVAS-Static	76.9	62.4	65.9	66.1	57.9	1497.6	63.1	56.6	84.0	31.3	99.1%
DOVAS	**78.9**	**63.0**	**69.2**	65.8	**59.4**	**1511.1**	**65.2**	**59.3**	86.7	30.7	**101.1%**

optimizer-aligned allocation guided by answer and visual counterfactual objectives. The main comparison evaluates this complete design.

DOVAS exceeds full-data aggregate performance on both datasets under the stated online compute cap, with a larger relative gain on Vision-Flan-186K. DOVAS-Static freezes the initialized region distribution throughout training. Its aggregate performance exceeds random selection, supporting the evidence scores as an allocation prior. DOVAS further improves over this fixed prior on both datasets. These results support the evaluated dynamic allocation design, while the objective and estimator ablations distinguish the signals responsible for its gains.

4.3 Generalization Ability

To examine whether DOVAS depends on a particular model architecture or capability level, we further evaluate it on Qwen2.5-VL-7B and Qwen3.5-4B. Both models are fine-tuned on LLaVA-665K under the same experimental setting used in the main experiments. Together with the primary LLaVA-v1.5-7B results, this evaluation covers different model families and capability levels. We apply the same partitioning and allocation procedure without introducing model-specific selection rules. As shown in Table 3, random sampling loses overall performance under the reduced budget, whereas DOVAS improves both models over their respective full-data baselines using only 50% of the online compute budget used for full-data training. The improvement is also observed across most individual tasks rather than being driven by a single benchmark. These results show that DOVAS is not tied to LLaVA or a specific model capacity, and the online utilities adapt the sampling distribution to the current target model and optimizer.

4.4 Ablation Study

4.4.1 Incremental Contribution of Visual Counterfactual Objectives

Visual counterfactual objectives improve over answer-only dynamic allocation. Table 4 compares three dynamic allocators that use the same fixed evidence regions and optimizer-aligned update estimation, but differ in the proxy objectives that guide allocation. The answer-only baseline uses A to favor updates predicted to reduce real-image answer loss. We then add V, which compares the

Table 2: Overall performance on Vision-Flan-186K using LLaVA-v1.5-7B with full-parameter fine-tuning. All selection methods use a matched online compute budget equal to 50% of that used for full-data training, while Full Data uses the entire training set. Offline annotation and preparation are excluded. Rel. is the mean of benchmark-wise performance normalized by Full Data. The best result in each column is shown in bold.

Method	VQAv2	GQA	LLaVA-W	SQA-I	TextVQA	MME-P	MMB-EN	MMB-CN	POPE	MM-Vet	Rel.
Full Data	70.4	47.3	38.6	57.8	40.5	1272.6	55.8	50.7	86.3	28.9	100.0%
Random	67.5	43.8	36.2	63.9	40.8	1280.9	44.9	45.5	83.3	29.7	96.4%
Self-Filter	66.1	44.6	34.9	61.1	45.9	1310.5	45.4	45.4	81.8	26.7	95.9%
EL2N	66.1	44.2	34.3	64.6	47.0	1312.4	46.8	39.9	80.0	28.5	96.1%
TypiClust	67.0	44.4	33.9	61.7	40.8	1249.6	34.5	47.7	84.4	30.5	94.2%
IFD	66.8	43.9	33.4	62.2	46.4	1265.5	33.2	43.7	86.7	29.5	94.4%
PreSel	64.5	41.9	42.9	61.9	42.1	1244.2	50.9	48.1	83.5	31.1	99.1%
XMAS	67.1	51.2	37.5	61.1	45.4	1189.7	53.4	46.7	76.9	24.8	97.5%
COINCIDE	67.4	45.4	37.9	65.8	35.5	1212.4	51.3	50.6	84.8	29.2	97.8%
CLIP-Score	63.6	43.4	33.9	64.9	44.3	1157.4	41.2	46.9	82.9	29.2	94.6%
CoIDO	66.5	47.2	41.1	**66.9**	45.3	1297.1	52.5	48.5	84.1	29.9	102.1%
ICONS	65.4	48.7	37.6	62.1	**47.3**	1254.2	53.1	46.5	81.9	29.7	100.1%
ScalSelect	68.3	47.6	37.7	58.2	41.4	1269.7	55.4	51.1	85.2	28.7	99.6%
DOSE	67.4	46.6	37.6	55.3	40.2	1274.2	54.5	51.7	84.7	29.5	98.7%
Once-For-All	71.2	49.4	39.7	58.7	42.4	1298.5	57.4	52.1	91.5	29.8	103.1%
TIVE	69.8	46.5	36.9	57.5	41.4	1258.6	56.3	49.7	82.9	29.9	99.2%
OPUS	72.5	49.5	42.0	60.7	43.4	1337.2	58.1	53.1	90.3	29.7	105.0%
VIG	73.1	49.8	41.1	59.7	43.7	1341.5	58.4	52.6	92.2	30.1	105.2%
VisNec	69.8	**53.0**	39.4	62.6	42.3	**1384.9**	60.1	51.2	85.2	31.0	105.0%
DOVAS-Static	74.2	51.5	42.8	60.9	43.5	1356.1	57.9	55.1	91.5	30.6	106.9%
DOVAS	**76.5**	52.0	**44.5**	61.7	45.1	1378.1	**61.1**	**55.7**	**94.0**	**31.4**	**109.7%**

Table 3: Cross-model generalization on LLaVA-665K with full-parameter fine-tuning. The best result within each model block is shown in bold.

Model	Method	VQAv2	GQA	LLaVA-W	SQA-I	TextVQA	MME-P	MMB-EN	MMB-CN	POPE	MM-Vet	Rel.
Qwen2.5-VL-7B	Full	83.5	64.3	78.3	77.5	79.2	1631.2	82.4	78.7	87.6	52.3	100.0%
	Random	82.5	62.1	77.6	76.6	79.3	1558.4	80.3	75.1	84.2	48.4	97.1%
	DOVAS	**83.8**	**65.1**	**80.5**	**78.5**	**81.8**	**1652.3**	**85.5**	**81.2**	**87.8**	**56.4**	**102.5%**
Qwen3.5-4B	Full	81.4	62.1	85.1	87.4	81.9	1731.2	84.8	83.6	**87.9**	**59.1**	100.0%
	Random	78.2	60.1	82.2	84.9	78.1	1681.5	82.9	82.1	87.2	56.7	97.0%
	DOVAS	**82.7**	**62.5**	**87.2**	**91.0**	**84.4**	**1754.8**	**86.1**	**84.9**	86.2	58.3	**101.3%**

real image with the blind condition, followed by S, which compares the real image with a matched negative image. Here A, V, and S abbreviate the regional utilities $A_b^{(q)}$, $V_b^{(q)}$, and $S_b^{(q)}$. Answer-only dynamic allocation remains below DOVAS-Static in aggregate on Vision-Flan, whereas adding the visual counterfactual objectives reverses this ordering. Within this framework, changing the distribution alone is insufficient; the objectives guiding allocation also matter. Appendices F.2, F.4, and F.5 examine the partition, the average fixed mixture, and the raw-gradient alternative; diagnostics and repeated runs appear in Appendices F.3 and F.6.

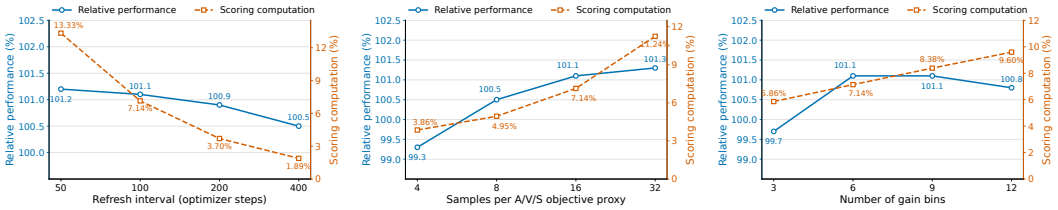
4.4.2 Performance and Computational Cost Trade-off

We study the accuracy and computational cost trade-off of the main DOVAS hyperparameters using LLaVA-v1.5-7B on LLaVA-665K. In Figure 3, the left axis reports performance relative to full-data training, and the right axis reports online-selection FLOPs as a percentage of online compute, $100C_{\text{online selection}}/C_{\text{total}}$. Parameter training and online selection count toward the budget, which is capped at 50% of the full-data baseline training compute. Offline annotation and preparation are excluded; the accounting boundary is specified in Appendix D.

Refresh interval. A shorter refresh interval updates the region probabilities more frequently and follows changes in the model and optimizer more closely. This improves sampling decisions, but

Table 4: **Visual counterfactual objectives add value beyond answer-only dynamic allocation.** Results on Vision-Flan-186K using LLaVA-v1.5-7B. All rows use dynamic allocation with the same fixed evidence regions and optimizer-aligned update estimation. Rows cumulatively add the real–blind (V) and real–negative (S) objectives to the answer-only baseline (A); the final row is full DOVAS. The best result in each column is shown in bold.

Allocation objectives	VQAv2	GQA	LLaVA-W	SQA-I	TextVQA	MME-P	MMB-EN	MMB-CN	POPE	MM-Vet	Rel.
Answer-only (A)	74.4	50.1	41.9	60.7	44.0	1351.8	57.1	54.4	92.4	29.9	106.0%
+ Real vs. blind ($A + V$)	75.8	51.4	43.7	60.4	44.1	1361.6	59.7	54.9	92.8	30.9	108.0%
+ Real vs. negative ($A + V + S$; DOVAS)	**76.5**	**52.0**	**44.5**	**61.7**	**45.1**	**1378.1**	**61.1**	**55.7**	**94.0**	**31.4**	**109.7%**



(a) Allocation refresh interval.

(b) Proxy mini-batch size per refresh.

(c) Evidence-region granularity.

Figure 3: Hyperparameter sensitivity and scoring cost of DOVAS on LLaVA-665K using LLaVA-v1.5-7B. The blue curve reports performance relative to full-data training, and the orange curve reports online-selection FLOPs as a percentage of online compute, $100C_{\text{online selection}}/C_{\text{total}}$.

frequent scoring also incurs additional computation. We therefore use an interval of 100 optimizer steps, which retains strong performance at a moderate overhead.

Proxy mini-batch size. We vary the number of examples drawn per objective at each allocation refresh while keeping the fixed proxy pools $\mathcal{P}_a$, $\mathcal{P}_v$, and $\mathcal{P}_s$ unchanged. Larger proxy mini-batches give more reliable gradient estimates for the A/V/S utilities. The gain largely saturates at 16 samples per objective per refresh, whereas the online-selection cost continues to increase. We therefore use $|\mathcal{Q}_x^{(q)}| = 16$ for each active objective in the remaining experiments. The horizontal axis in Figure 3b denotes this per-refresh mini-batch size, not the number of samples reserved in a proxy pool.

Number of evidence bins. The number of bins controls the granularity of the fixed data partition. Too few bins merge samples with different evidence levels, while too many bins fragment the data and make regional estimates less stable. Six bins match the best observed performance with lower overhead than finer partitions and are used as the default. Overall, performance remains stable around these choices, showing that DOVAS is not sensitive to a narrow hyperparameter setting.

5 Conclusion

Data selected before training need not remain equally useful as a multimodal model learns. Dynamic selection can respond to these changes, but must balance the cost of reassessment with the relevance of its scoring objective. In particular, answer-loss reduction alone does not establish whether an update improves the use of visual evidence. We introduced DOVAS to guide selection through both answer and visual counterfactual objectives. It organizes data into fixed evidence regions within semantic clusters, then uses the current optimizer state to estimate how regional updates affect predictions with correct, missing, and incorrect images. Periodic regional probes allow the sampling distribution to adapt without rescoring every example or rebuilding the partition. Across the evaluated datasets and models, DOVAS matches or exceeds full-data performance on average under an online compute cap equal to half the baseline training compute. These results suggest that multimodal data value should not be treated as a fixed property of a sample, but should also reflect when and how that sample contributes to the current optimization process.

6 AI USE STATEMENT

We only use AI to aid or polish writing

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization. *Text Reading, and Beyond*, 2(1):1, 2023.

Mayee Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. Skill-it! a data-driven skills framework for understanding and training language models. *Advances in Neural Information Processing Systems*, 36:36000–36040, 2023.

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, 2024.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

Mingkang Dong, Hongyi Cai, Xiwen Lei, Jie Li, Tao Zhang, and Muxin Pu. Once-for-all: A train-once and select-anytime framework for multimodal instruction tuning. *arXiv preprint arXiv:2605.26761*, 2026a.

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. *arXiv preprint arXiv:2603.01195*, 2026b.

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

Hongzhe Huang, Jiang Liu, Zhewen Yu, Li Cai, Dian Jiao, Wenqiao Zhang, Siliang Tang, Juncheng Li, Hao Jiang, Haoyuan Li, et al. Align2llava: Cascaded human and large language model preference alignment for multi-modal instruction curation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 8759–8781, 2025.

Jaewoo Lee, Boyang Li, and Sung Ju Hwang. Concept-skill transferability-based data selection for large vision-language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5060–5080, 2024.

Seulbi Lee and Sangheum Hwang. Selective training for large vision language models via visual information gain. *arXiv e-prints*, pp. arXiv–2602, 2026.

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Joshua Adrian Cahyono, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(9):7543–7557, 2025.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PmLR, 2023.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296. IEEE, 2024.

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.

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

Nilay Naharas, Dang Nguyen, Nesihan Bulut, Mohammadhossein Bateni, Vahab Mirrokni, and Baharan Mirzasoleiman. Data selection for fine-tuning vision language models via cross modal alignment trajectories. *arXiv preprint arXiv:2510.01454*, 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Bardia Safaei, Faizan Siddiqui, Jiacong Xu, Vishal M Patel, and Shao-Yuan Lo. Filter images first, generate instructions later: Pre-instruction data selection for visual instruction tuning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14247–14256. IEEE, 2025.

Shaobo Wang, Xuan Ouyang, Tianyi Xu, Yuzheng Hu, Jialin Liu, Guo Chen, Tianyu Zhang, Junhao Zheng, Kexin Yang, Xingzhang Ren, et al. Opus: towards efficient and principled data selection in large language model pre-training in every iteration. *arXiv preprint arXiv:2602.05400*, 2026.

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*, 2(6), 2023.

Biao Wu, Yiwu Zhong, Meng Fang, and Ling Chen. Dose: Data selection for multi-modal llms via off-the-shelf models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 897–911, 2026a.

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*, 2026b.

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. *arXiv preprint arXiv:2402.04333*, 2024.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36: 34201–34227, 2023.

Zhiyang Xu, Chao Feng, Rulin Shao, Trevor Ashby, Ying Shen, Di Jin, Yu Cheng, Qifan Wang, and Lifu Huang. Vision-flan: Scaling human-labeled tasks in visual instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 15271–15342, 2024.

Yichen Yan, Ming Zhong, Qi Zhu, Xiaoling Gu, Jinpeng Chen, and Huan Li. Coido: Efficient data selection for visual instruction tuning via coupled importance-diversity optimization. *Advances in Neural Information Processing Systems*, 38:167045–167073, 2026.

Bo Zhao, Boya Wu, Muyang He, and Tiejun Huang. Svit: Scaling up visual instruction tuning. *arXiv preprint arXiv:2307.04087*, 2023.

Deyao Zhu, Xiaoqian Shen, Xiang Li, Mohamed Elhoseiny, et al. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *International Conference on Learning Representations*, volume 2024, pp. 18378–18394, 2024.

A Optimization Interpretation of Regional Allocation

This section gives an optimization interpretation of the regional utilities and dynamic allocation used by DOVAS. We analyze the allocation problem at a single refresh while treating the measured utilities as fixed. The result characterizes the allocation objective gap under the given utilities; it does not compare the training trajectories induced by different allocation policies.

A.1 One-Step Regional Update Utility

Let $\mathcal{I} = \{0, \dots, M\}$ be the set of evidence regions. At refresh q, DOVAS draws a probe batch $\mathcal{C}_b^{(q)}$ from each region b and computes

$$g_b^{(q)} = \nabla_\theta \mathcal{L}_{\mathrm{MM}}(\theta_q; \mathcal{C}_b^{(q)}). \tag{21}$$

At refresh q, the fixed proxy pools supply mini-batches $\mathcal{Q}_a^{(q)}$, $\mathcal{Q}_v^{(q)}$, and $\mathcal{Q}_s^{(q)}$. The three stochastic proxy gradients are

$$g_x^{(q)} = \nabla_\theta \widehat{J}_x^{(q)}(\theta_q), \qquad x \in \{a, v, s\}, \tag{22}$$

where $\widehat{J}_x^{(q)}$ is the mini-batch objective defined in Equation 13. We condition on these sampled mini-batches when interpreting the measured utilities below. Rather than assuming that the optimizer is a fixed linear preconditioner, DOVAS applies its complete update rule to a copy of the current optimizer state:

$$(\theta_{q,b}^+, s_{q,b}^+) = \mathrm{Opt}(\theta_q, s_q, g_b^{(q)}), \qquad u_b^{(q)} = \theta_q - \theta_{q,b}^+. \tag{23}$$

Thus, $u_b^{(q)}$ already includes the current learning rates, parameter-group scales, optimizer moments, and weight decay. When the optimizer admits a local linear approximation, the update takes the form $u_b^{(q)} \approx P_q g_b^{(q)}$, but the analysis below does not require this approximation.

For any sampled proxy objective $\widehat{J}^{(q)}$ with a locally Lipschitz gradient, a first-order expansion around θ_q gives

$$\widehat{J}^{(q)}(\theta_{q,b}^+) = \widehat{J}^{(q)}(\theta_q) - \left\langle u_b^{(q)}, \nabla_\theta \widehat{J}^{(q)}(\theta_q) \right\rangle + \mathcal{O}(\|u_b^{(q)}\|^2). \tag{24}$$

This motivates the three regional utilities

$$A_b^{(q)} = \left\langle u_b^{(q)}, g_a^{(q)} \right\rangle, \quad V_b^{(q)} = \left\langle u_b^{(q)}, g_v^{(q)} \right\rangle, \quad S_b^{(q)} = \left\langle u_b^{(q)}, g_s^{(q)} \right\rangle. \tag{25}$$

A positive value predicts a local reduction in the corresponding sampled proxy objective, up to the second-order remainder. Prediction of improvement over the full proxy pool additionally depends on mini-batch gradient estimation error. In particular, $A_b^{(q)}$ measures the predicted reduction in real-image answer loss. Since J_v and J_s are defined as the real-image loss minus the blind and negative-image losses, respectively, reducing them increases the counterfactual-minus-real loss gaps. This describes proxy losses rather than guaranteeing improved visual capability. Appendix F.3 examines the correct-image and counterfactual terms separately.

A.2 Entropy-Regularized Region Allocation

After cross-region standardization and exponential moving-average smoothing, the combined utility used in the main method is

$$U_b^{(q)} = \widehat{A}_b^{(q)} + \widehat{V}_b^{(q)} + \widehat{S}_b^{(q)}. \tag{26}$$

Let $K = |\mathcal{I}| = M + 1$ and $\Delta_K = \{\pi \in \mathbb{R}_+^K : \sum_{b \in \mathcal{I}} \pi_b = 1\}$. For a fixed refresh, the DOVAS allocation is the solution to

$$\max_{\pi \in \Delta_K} \quad F_q(\pi) = \sum_{b \in \mathcal{I}} \pi_b U_b^{(q)} + \tau H(\pi), \qquad H(\pi) = -\sum_{b \in \mathcal{I}} \pi_b \log \pi_b. \tag{27}$$

The first term favors regions with higher combined measured utility for the proxy objectives. The entropy term controls concentration and is, up to an additive constant, equivalent to a KL penalty toward the uniform distribution.

To obtain the solution, introduce a multiplier ν for probability normalization. For an interior point, stationarity of

$$\mathcal{G}(\pi, \nu) = \sum_b \pi_b U_b^{(q)} - \tau \sum_b \pi_b \log \pi_b + \nu \left(\sum_b \pi_b - 1 \right) \tag{28}$$

requires

$$U_b^{(q)} - \tau(\log \pi_b + 1) + \nu = 0. \tag{29}$$

Normalizing the resulting exponential weights gives

$$\boxed{\pi_{q,b}^* = \frac{\exp(U_b^{(q)}/\tau)}{\sum_{j \in \mathcal{I}} \exp(U_j^{(q)}/\tau)} = \mathrm{softmax}_b\Big(U^{(q)}/\tau\Big).} \tag{30}$$

For $\tau > 0$, entropy is strictly concave on the simplex, while the utility term is linear. Hence, F_q is strictly concave and Equation 30 is its unique global maximizer. The softmax also assigns positive probability to every region for finite utilities. Semantic coverage is preserved separately through the cluster shares ρ_c used within each region.

This formulation matches Equation 19 in the main text. In particular, the initial distribution π_0 is used only for the first training block and is not a KL prior after the first refresh. The implemented combined utility also assigns equal weight to the three standardized components; it does not introduce constraint multipliers or dual updates.

A.3 Allocation Suboptimality under Measured Utilities

The allocation objective characterizes the objective gap when a fixed or previously optimal distribution is evaluated under the current measured utilities. Define

$$Z_q = \sum_{j \in \mathcal{I}} \exp(U_j^{(q)}/\tau), \qquad \log \pi_{q,b}^* = \frac{U_b^{(q)}}{\tau} - \log Z_q. \tag{31}$$

For any $\pi \in \Delta_K$,

$$\begin{aligned} \tau D_{\mathrm{KL}}(\pi \| \pi_q^*) &= \tau \sum_b \pi_b \log \frac{\pi_b}{\pi_{q,b}^*} \\ &= \tau \log Z_q - F_q(\pi). \end{aligned} \tag{32}$$

Evaluating this identity at π_q^* gives $F_q(\pi_q^*) = \tau \log Z_q$, and therefore

$$\boxed{F_q(\pi_q^*) - F_q(\pi) = \tau D_{\mathrm{KL}}(\pi \| \pi_q^*).} \tag{33}$$

If the allocation from the preceding refresh is retained, its current allocation objective gap is

$$F_q(\pi_q^*) - F_q(\pi_{q-1}^*) = \tau D_{\mathrm{KL}}(\pi_{q-1}^* \| \pi_q^*). \tag{34}$$

The gap is zero when the relative regional utility profile is unchanged and is strictly positive whenever the old and current optimal allocations differ. For any fixed allocation π^{static}, summing over refreshes gives

$$\boxed{\sum_q \left[F_q(\pi_q^*) - F_q(\pi^{\mathrm{static}}) \right] = \tau \sum_q D_{\mathrm{KL}}(\pi^{\mathrm{static}} \| \pi_q^*).} \tag{35}$$

This identity sums objective gaps over a given sequence of measured utilities. The fixed and adaptive distributions are evaluated against that same sequence; separately training them would generally produce different model states and utilities. The result therefore explains the allocation rule at each refresh, without proving that its training trajectory outperforms a static policy, converges globally, or improves final benchmark performance.

Table 5: Target-model training configurations. Full Data is trained for one epoch; all methods within a model block use the same optimizer, effective batch size, and learning-rate schedule.

Target model	Optimizer	Peak LR	Effective batch	Schedule
LLaVA-v1.5-7B	AdamW	1×10^{-5}	16	3% warm-up + cosine to 5×10^{-6}
Qwen2.5-VL-7B	AdamW	1×10^{-5}	16	3% warm-up + cosine to 5×10^{-6}
Qwen3.5-4B	AdamW	1×10^{-5}	16	3% warm-up + cosine to 5×10^{-6}

B Implementation Details

Training configuration. For every dataset–model pair, the Full Data baseline is trained for one epoch over the complete eligible training pool. Text-only records and the common held-out proxy samples are removed before training, and the resulting multimodal pool is shared by Full Data and all selection methods. We train the language model and multimodal connector in BF16 using AdamW, keep the visual encoder frozen, and clip the gradient norm to 1.0. Table 5 summarizes the model-specific configurations. The effective batch size denotes the global number of examples after gradient accumulation. Each learning-rate schedule uses linear warm-up over the first 3% of parameter-training updates, followed by cosine decay to one half of the peak learning rate. The scheduler advances only after a parameter update; proxy and regional-probe computations do not advance it. For budgeted methods, the schedule is rescaled to the number of parameter updates allowed by the remaining online compute budget. Random sampling, static selection, and the OPUS protocol are specified in Appendix C.

Dynamic-allocation configuration. We construct five equal-frequency positive-evidence regions and one additional region for samples that do not satisfy the real-image condition, giving six regions in total. DOVAS refreshes the allocation every $R = 100$ optimizer steps. At each refresh, we draw a semantically stratified probe batch of eight examples from every region, corresponding to 48 regional probe examples in total. For each active objective $x \in \{a, v, s\}$, we draw a fresh mini-batch $\mathcal{Q}_x^{(q)}$ of 16 examples from its fixed proxy pool $\mathcal{P}_x$ and compute the stochastic gradient in Equation 13. These mini-batches are shared across the regional utility evaluations within a refresh; they are drawn again at the next refresh, while pool membership remains fixed.

We set the initial-distribution temperature τ_0 and the dynamic allocation temperature τ to 1.0. The standardized regional utilities are smoothed with an EMA factor of $\beta = 0.9$. All three objectives have equal weights. The value of τ is fixed using a pre-training allocation-stability criterion without consulting downstream evaluation results.

Semantic clustering and proxy construction. We remove the image placeholder from the first user instruction and encode the remaining text using Qwen3.5-4B. The embeddings are partitioned into 20 semantic clusters using MiniBatchKMeans. Evidence scores and region boundaries are computed within each semantic cluster to avoid directly comparing samples from unrelated task types.

We reserve 1.5% of every semantic-cluster–region cell as a common held-out proxy pool. Proxy quotas are proportional to the original sizes of these cells. The answer proxy pool $\mathcal{P}_a$ retains samples with the largest answer-compatibility scores q_i^a. The visual proxy pool $\mathcal{P}_v$ retains samples with positive real-versus-blind gain and large q_i^v. The specificity proxy pool $\mathcal{P}_s$ contains at most 5,000 independently selected targets with verified negative images. For each target, we retrieve semantically similar candidates from the same instruction cluster while excluding the original image and candidates with identical or strongly overlapping answers. A pair is retained only if swapping the two images increases the corresponding response loss in both directions by more than 0.01. Target examples and negative images are unique within $\mathcal{P}_s$. All proxy samples are excluded from parameter training for every compared method. These rules determine the fixed pool membership and sizes; the 16-example setting controls only the mini-batch drawn from each active objective pool at each refresh.

C BASELINE PROTOCOL UNDER THE ONLINE COMPUTE BUDGET

This section details the baseline configurations used in Tables 1 and 2. Random sampling, static selection, and the multimodal adaptation of OPUS use the same online compute cap. The 50% setting specifies computation relative to full-data training, rather than a common fraction of retained examples.

C.1 COMMON TRAINING CONDITIONS AND ACCOUNTING BOUNDARY

Let $\mathcal{D}_{\mathrm{tr}}$ denote the eligible multimodal instruction pool after removing text-only records and the common held-out examples. The exclusions belong to the shared dataset manifest. All methods for a dataset–model pair use the same starting checkpoint, image preprocessing, conversation template, sequence-length limit, trainable parameters, AdamW configuration, and global update batch size of 16. The visual encoder is frozen, and the language model and multimodal connector are trained, following Appendix B.

Full Data makes one pass over $\mathcal{D}_{\mathrm{tr}}$, defining C_{full}. Each budgeted run obeys

$$C_{\mathrm{training}} + C_{\mathrm{online\ selection}} \leq \alpha C_{\mathrm{full}}, \qquad \alpha = 0.5 \quad \text{in the reference configuration.} \tag{36}$$

The accounting boundary is the same as in Appendix D. Selection cost includes candidate and proxy forward/backward passes, preconditioning, gradient projections, and selection arithmetic, including work on rejected candidates. Offline preparation is accounted for separately. All baseline scores in the comparison are obtained under this common training protocol.

C.2 RANDOM SAMPLING AND STATIC SELECTION

Random in the main tables samples training examples at random from the common eligible pool $\mathcal{D}_{\mathrm{tr}}$, without replacement. It traverses a random ordering of this pool until the next update would exceed the online compute cap. It uses the same training configuration as the other budgeted methods.

Static methods follow the scoring, clustering, and diversity rules of their respective baseline implementations. We expand the candidate set under these rules to supply enough distinct examples for one training pass within $0.5C_{\mathrm{full}}$. The selected pool is fixed before parameter training, and its examples are used without replacement. Budget matching therefore uses a larger candidate set rather than repeated epochs over a small subset. The retained fraction is determined by the cost of this single pass, with any online-selection cost charged to the same cap; it is not selected using downstream test scores. Retained sample counts depend on input lengths and selection costs. Training stops at the last complete update allowed by the cap, or when eligible examples are exhausted.

With the global batch size of 16, a run completing K_m full parameter updates visits $16K_m$ distinct training examples under this sampling protocol. Its realized coverage is $16K_m/|\mathcal{D}_{\mathrm{tr}}|$. A 50% compute cap does not imply that exactly half of the examples are visited. These single-pass rules apply to Random and the static subset baselines.

DOVAS-Static instead fixes the initialized sampling mixture over the eligible pool, using the same evidence regions as DOVAS. DOVAS updates this mixture during training. Both use the same online cap. For all budgeted methods, executed FLOPs determine stopping as specified in Appendix D; the learning-rate schedule advances only with parameter updates, following Appendix B.

C.3 VISNEC AND VIG SELECTION RULES

For offline visual scoring, VisNec and VIG use the same Qwen3.5-9B checkpoint as DOVAS. The comparison therefore holds the scoring model fixed while retaining each method's selection criterion. This shared offline scorer is separate from the target model used for training and online proxy gradients.

VisNec computes its two-condition scores with this shared scorer and uses 20 instruction clusters (Dong et al., 2026b). Only examples with positive visual-necessity scores are eligible for selection. These rules define its selection criterion, while the common online budget controls training computation.

VIG preserves token selection (Lee & Hwang, 2026): the full answer is supplied as context, but only answer tokens meeting the sample-selection threshold contribute to the loss. Its visual-absence condition uses Gaussian blur, and its scores are computed with the shared Qwen3.5-9B scorer. Masking answer-token losses does not remove their contextual computation or automatically reduce transformer FLOPs; accounting follows the operations actually executed.

C.4 OPUS Inputs, Proxy Objective, and Candidate Selection

OPUS adapts the selection algorithm of Wang et al. (2026) to visual instruction tuning. Each candidate is a complete (v_i, x_i, y_i) example. Scoring uses per-example mean answer-token negative log-likelihood conditioned on the real image and instruction. It uses at most the first 512 answer tokens while retaining the complete image-token block and the instruction under the common preprocessing limit. Selected examples receive the ordinary training loss over all answer tokens allowed by the common sequence limit. Candidate and proxy gradients propagate through the multimodal connector and language model; masking image and instruction labels does not detach their activations.

The proxy pool is DOVAS's fixed answer pool $\mathcal{P}_a$, with real-image answer loss as its objective. This shared, training-derived pool replaces the benchmark-retrieved Bench-Proxy of the original pipeline. OPUS thus specifies a common-proxy multimodal selector, rather than the original benchmark-informed OPUS pipeline.

A single uniformly sampled training update initializes the AdamW moments and is charged to the budget. At each subsequent parameter update, the selector draws 32 distinct candidates from a reshuffled training stream and a fresh 16-example proxy batch, then selects 16 candidates without replacement. Rejected candidates are consumed from the stream and become eligible again at its next reshuffle. Selection uses the current AdamW state, the frozen-preconditioner approximation, and OPUS's sequential proxy-alignment score with its accumulated within-batch redundancy penalty. The proxy direction is unpreconditioned. The effective candidate displacement includes the learning rate exactly once.

Selection-only statistics are discarded before the ordinary clipped training update and do not alter the stored optimizer state. OPUS uses no evidence regions, regional EMA, or visual counterfactual objectives. It is therefore operationally distinct from DOVAS's answer-only regional ablation. Candidate selection and the resulting training update are both subject to the remaining online budget.

C.5 Projection, Score Calibration, and Reproducibility

OPUS uses fixed, layer-specific CountSketch maps of dimension 8192 with projection seed 42. Candidate and proxy sketches within a layer share the same map. Linear-layer gradients are accumulated over token positions before projection; an entire sequence is not treated as a single rank-one gradient. Matrix contributions are processed in layer/chunk order using activation and backpropagation factors. Other trainable parameter types contribute their directly computed per-example gradients, and tied parameters are counted once. Inner products and score statistics accumulate in FP32. Projection and preconditioning are charged to online selection.

At each sequential draw, the current redundancy-adjusted scores over the remaining candidates are standardized by their mean and standard deviation, with a 10^{-12} denominator floor. Boltzmann sampling uses temperature 0.9; zero-variance scores yield uniform sampling. This score calibration is an explicit OPUS adaptation for variable-length multimodal losses.

The machine-readable record format includes exact checkpoint and scorer identifiers, code revision, trainable parameter names, training and proxy manifests, selected-pool sizes, random seeds, update count, distinct examples visited, effective passes, processed and supervised token counts, and realized training and selection FLOPs. Runtime, peak memory, offline preparation, and configuration-search costs are separate fields.

D TRAINING AND ONLINE SELECTION COMPUTE

We measure the online computation of one instruction-tuning run as

$$C_{\text{total}} = C_{\text{training}} + C_{\text{online selection}}. \tag{37}$$

Here C_{total} denotes the total within this online accounting boundary. Both terms use the same floating-point operation (FLOP) convention, with costs determined by the target model, input shapes, and operations executed. Token counts alone do not account for differences in visual processing or forward and backward computation.

Offline annotation and preparation. Offline annotation and preparation are outside the reported compute budget for all methods. For DOVAS, these operations include instruction encoding and clustering, Qwen3.5-9B scoring under the real, blind, and negative-image conditions, negative-image retrieval and verification, and construction of the evidence regions and proxy pools. The prepared regions, annotations, and proxy pools are inputs to the online training run. Method-specific offline scoring and selector preparation for baselines follow the same exclusion.

Training computation. C_{training} includes the forward and backward computation and optimizer updates used to train the target model. It includes forward passes through the frozen visual encoder and activation recomputation during backpropagation. Probe and proxy evaluations used only to guide selection belong to the online-selection term.

Online-selection computation. $C_{\text{online selection}}$ includes all allocation refreshes: stochastic proxy-gradient evaluation on the sampled mini-batches $\mathcal{Q}_x^{(q)}$ under the required real, blind, and negative conditions; regional probe gradients; virtual optimizer-update arithmetic; utility evaluation; and normalization, smoothing, and probability updates. Shared computations within a refresh are counted once. State copying and memory traffic affect runtime but do not themselves contribute FLOPs.

Budget normalization and stopping. Let C_{full} denote the parameter-training compute of the corresponding full-data baseline on the same dataset and target model. A budget ratio α specifies the online compute cap

$$B = \alpha C_{\text{full}}, \qquad C_{\text{total}} \le B. \tag{38}$$

The default 50% budget uses $\alpha = 0.5$ and includes both terms in Equation 37. Every method's online selection is charged against its cap; offline annotation and preparation are excluded. Training starts from the pretrained checkpoints; their original pretraining and downstream benchmark evaluation are also outside this boundary for all methods.

After offline preparation, the online compute counter starts at zero. The budget covers at least one parameter-training update. Before executing an update or an allocation refresh, its FLOP cost is determined for the scheduled inputs and checked against the remaining budget. The run stops before the next scheduled operation exceeds B, leaving any insufficient residual budget unused. Algorithm 1 denotes these operation costs by c_{train} and $c_{\text{selection}}$, respectively.

The online-selection overhead is $100C_{\text{online selection}}/C_{\text{total}}$ percent. Under a fixed online budget, increasing selection computation reduces the computation available for parameter training. Comparisons across offline scorer sizes use the same online cap; offline scoring is not deducted from it.

D.1 COMPUTE BREAKDOWN AT THE 50% CAP

Table 6 gives an equal-cost reference estimate for LLaVA-v1.5-7B on LLaVA-665K under the default allocation settings. With an effective training batch size of 16, each interval of 100 optimizer steps processes 1,600 training examples, counting repeated visits. One allocation refresh evaluates $6 \times 8 = 48$ regional probe examples under the real-image condition, 16 answer-objective conditions, $2 \times 16 = 32$ visual-objective conditions, and $2 \times 16 = 32$ specificity-objective conditions. The total is therefore $48 + 16 + 32 + 32 = 128$ sample–condition evaluations. The A/V/S gradients are shared across regions, so their condition counts are not multiplied by the number of regions.

Table 6: **Reference compute breakdown at the 50% online budget.** LLaVA-v1.5-7B on LLaVA-665K with the default allocation settings. Values use the full-image, equal-cost approximation in Equation 39, rather than measured run-level FLOPs. The selection share is a conservative reference for sample–condition evaluations.

Component	Reference share of DOVAS online compute (%)	Reference share of Full Data training compute (%)
Parameter training	92.59	46.30
Online selection	7.41	3.70
Total within budget	100.00	50.00
Offline annotation/preparation	Excluded	Excluded

Charging every sample–condition evaluation as a full real-image evaluation gives an online-selection share of

$$h_{\text{ref}} = \frac{128}{1{,}600 + 128} \approx 7.41\%. \tag{39}$$

We use this as a conservative upper reference for the sample–condition evaluation cost under this approximation. In the blind condition, image placeholders are replaced and the expanded visual-token block is omitted, so those evaluations cost less than full real-image evaluations. This reduces the evaluation cost relative to the reference calculation. The 7.41% value is neither a measured run-level share nor an unconditional bound on all selection FLOPs: virtual updates and other selection arithmetic also contribute. We count these operations in $C_{\text{online selection}}$ using the FLOP convention defined above.

For a run with selection share $h = C_{\text{online selection}}/C_{\text{total}}$, let $f = C_{\text{total}}/C_{\text{full}} \leq 0.5$. Its training and selection costs relative to the full-data reference are $f(1-h)$ and fh, respectively.

E Algorithm Overview

Algorithm 1 summarizes the complete DOVAS pipeline. The offline stage constructs fixed evidence regions and proxy pools, while the online stage updates only the regional sampling distribution. All virtual optimizer steps are evaluated on copies of the current state and are discarded after the regional utilities are measured. The accumulated online compute includes parameter training and online selection as defined in Appendix D.

F Additional Experimental Results

F.1 Performance under Different Compute Budgets

Table 7 compares DOVAS with random sampling on LLaVA-665K at the reported online compute budgets. DOVAS achieves higher aggregate performance at each evaluated budget. At the lowest reported budget, however, the aggregate advantage is modest and the task-level comparison is mixed: DOVAS improves some benchmarks, ties another, and underperforms on others. At the higher budgets, its advantage extends across most benchmarks. The evidence therefore supports an aggregate benefit across the evaluated budgets, with more consistent task-level gains when more online computation is available.

Aggregate performance increases with budget for both methods. DOVAS improves most when moving from the lowest to the intermediate budget, with a smaller additional gain at the highest budget. Its advantage over random sampling widens at the intermediate budget and narrows at the highest budget while remaining positive. DOVAS exceeds the full-data baseline at the higher reported budgets, but remains below it at the lowest budget. These trends are limited to the budget settings shown in the table.

Algorithm 1 Dynamic Optimization-Aligned Data Selection (DOVAS)

Require: Training pool $\mathcal{D}$; fixed scorer O; target model θ and optimizer state s; number of positive-evidence bins M; refresh interval R; proxy mini-batch size m (default 16); temperatures τ_0, τ; EMA factor β; feasible online compute budget B

Ensure: Fine-tuned parameters θ

Offline initialization

Cluster instructions by semantic intent and assign each sample a cluster index c_i
Compute real-image and blind losses; retrieve and verify negative images within each cluster to obtain negative-image losses
Compute within-cluster scores q_i^a, q_i^v, q_i^s using Equation 4
Compute the real-image condition χ_i and evidence score $r_i = \chi_i q_i^a q_i^v q_i^s$
Construct fixed regions $\{\mathcal{D}_b\}_{b=0}^{M}$ using Equation 7
Set $\pi^{(0)} \leftarrow \mathrm{softmax}(\bar{\mathbf{r}}/\tau_0)$ and record cluster shares ρ_c
Build held-out proxy pools $\mathcal{P}_a, \mathcal{P}_v, \mathcal{P}_s$ and exclude them from parameter-training regions

Dynamic training

$q \leftarrow 0$ and $C_{\mathrm{total}} \leftarrow 0$
while $C_{\mathrm{total}} < B$ **do**
  for $k = 1$ to R **do**
    Draw region $b \sim \pi^{(q)}$ and form a batch $\mathcal{B}$ from $\mathcal{D}_b$ according to ρ_c
    Determine the training-update cost c_{train} for $\mathcal{B}$
    if $C_{\mathrm{total}} + c_{\mathrm{train}} > B$ **then**
      return θ
    end if
    Update (θ, s) with the standard instruction-tuning loss in Equation 2
    $C_{\mathrm{total}} \leftarrow C_{\mathrm{total}} + c_{\mathrm{train}}$
  end for
  Draw proxy mini-batches $\mathcal{Q}_x^{(q+1)} \subset \mathcal{P}_x$ of size m for each active objective x
  Draw stratified probes $\{\mathcal{C}_b^{(q+1)}\}_{b=0}^{M}$ and determine the full refresh cost $c_{\mathrm{selection}}$
  if $C_{\mathrm{total}} + c_{\mathrm{selection}} > B$ **then**
    return θ
  end if
  $q \leftarrow q + 1$; compute stochastic proxy gradients $g_x^{(q)}$ on $\mathcal{Q}_x^{(q)}$ using Equation 13
  for all $b \in \{0, \ldots, M\}$ **do**
    Compute $g_b^{(q)}$ from the stratified probe $\mathcal{C}_b^{(q)}$
    Apply one virtual optimizer step on a copy of (θ, s) to obtain $u_b^{(q)}$
    Compute $A_b^{(q)}, V_b^{(q)}, S_b^{(q)}$ using Equation 16
  end for
  Standardize the three utilities across regions and update their EMA values $\widehat{A}^{(q)}, \widehat{V}^{(q)}, \widehat{S}^{(q)}$
  $U_b^{(q)} \leftarrow \widehat{A}_b^{(q)} + \widehat{V}_b^{(q)} + \widehat{S}_b^{(q)}$ for every region b
  $\pi^{(q)} \leftarrow \mathrm{softmax}(U^{(q)}/\tau)$
  $C_{\mathrm{total}} \leftarrow C_{\mathrm{total}} + c_{\mathrm{selection}}$
end while
return θ

F.2 Effect of Region Construction and Scorer Scale

Table 8 separates two questions: whether evidence-based region construction helps online allocation, and how the offline scorer affects the resulting performance. Both comparisons use LLaVA-v1.5-7B on LLaVA-665K under the same online compute cap of $0.5C_{\mathrm{full}}$.

We replace the default evidence partition with either randomly assigned regions or regions defined only by instruction semantics. These two controls retain the same A/V/S proxy pools and online allocation mechanism as default DOVAS; only the region construction changes. Random partitioning here refers to assigning samples to regions, whereas Random in Table 1 refers to randomly selecting training data.

Table 7: Performance of DOVAS under different online compute budgets on LLaVA-665K using LLaVA-v1.5-7B, with random sampling evaluated at the same budgets. The online compute budget is measured relative to that used for full-data training. Rel. is the mean benchmark-wise performance normalized by the Full Data result in Table 1. The best result in each column within a budget is shown in bold, including ties. DOVAS rows are shaded.

Compute Budget	Method	VQAv2	GQA	LLaVA-W	SQA-I	TextVQA	MME-P	MMB-EN	MMB-CN	POPE	MM-Vet	Rel.
30%	Random	**74.1**	**56.2**	**60.8**	64.2	52.4	**1434.2**	60.6	**52.8**	83.8	29.6	93.7%
	DOVAS	73.2	55.4	59.0	**66.9**	**54.6**	1402.6	**60.7**	**52.8**	**84.6**	**31.0**	**94.3%**
50%	Random	74.5	56.3	58.8	**67.8**	56.2	1423.2	62.3	54.1	85.4	**31.7**	95.9%
	DOVAS	**78.9**	**63.0**	**69.2**	65.8	**59.4**	**1511.1**	**65.2**	**59.3**	**86.7**	30.7	**101.1%**
70%	Random	76.3	59.8	63.2	**68.1**	57.2	1472.5	63.1	56.3	86.1	29.1	97.6%
	DOVAS	**79.2**	**63.2**	**69.4**	66.9	**59.1**	**1519.8**	**65.8**	**59.7**	**87.1**	**31.8**	**102.0%**

Table 8: **Evidence-based region construction improves dynamic allocation.** Results on LLaVA-665K using LLaVA-v1.5-7B at an online compute cap of $0.5C_{\text{full}}$. The upper block replaces the default evidence partition with random or semantic-only regions while keeping the same A/V/S proxy pools and online allocation. The lower block uses evidence regions with different Qwen3.5 scorers; shading identifies the default scorer, Qwen3.5-9B. Rel. is the mean benchmark-wise performance normalized by the Full Data result in Table 1. The best result in each column is shown in bold.

Region construction or scorer	VQAv2	GQA	LLaVA-W	SQA-I	TextVQA	MME-P	MMB-EN	MMB-CN	POPE	MM-Vet	Rel.
Alternative region partitions											
Random partition	74.0	56.8	61.7	61.8	53.5	1398.6	58.8	52.5	81.5	26.1	91.8%
Semantic-only partition	74.5	56.2	61.5	61.2	54.0	1411.0	59.2	53.6	82.2	27.7	92.6%
Evidence-based partitions: Qwen3.5 scorer scale											
Qwen3.5-0.8B	76.2	59.8	64.7	62.8	55.5	1448.6	59.8	55.5	83.5	29.1	95.7%
Qwen3.5-2B	77.5	60.8	66.5	64.2	57.0	1477.2	62.8	56.9	85.1	29.4	97.9%
Qwen3.5-4B	78.2	62.0	67.8	65.4	58.5	1493.1	64.0	58.3	86.1	30.1	99.7%
Qwen3.5-9B	**78.9**	**63.0**	**69.2**	**65.8**	**59.4**	**1511.1**	**65.2**	**59.3**	**86.7**	**30.7**	**101.1%**

Both random and semantic-only partitions underperform the evidence-based partitions across the benchmarks. Even the evidence partition constructed with the smallest scorer exceeds both alternative partitions in aggregate. These results support organizing regions by visual evidence for the online allocator. In particular, semantic grouping alone does not recover the performance of the proposed partition, which distinguishes evidence levels within semantic clusters. This region-construction comparison complements the visual-objective ablation in Table 4: the former examines how samples are grouped for allocation, while the latter examines which proxy objectives guide the allocation.

The lower block varies the Qwen3.5 scorer used for the offline evidence scores, region boundaries, and proxy pools while retaining the target model, training pool, online compute cap, and allocation hyperparameters. Offline scoring cost is excluded for every scorer; parameter training and online selection share the same cap. This comparison evaluates how offline scoring quality affects the subsequent allocation and training under a fixed online budget. Performance improves across the tested scorer sizes, with diminishing gains in aggregate. Qwen3.5-9B achieves the strongest results among the tested scorers and is used as the default offline scorer; the smaller models serve as scale-ablation controls.

F.3 Update-Utility Validation and Independent Visual Diagnostics

Figure 4 connects update-level prediction, independent visual diagnostics, and the benchmark ablation. These measurements address complementary questions: whether predicted utility tracks observed improvement, whether larger visual loss gaps accompany better correct-image performance, and whether visual objectives improve downstream allocation.

Predicted versus observed update improvement. We compare optimizer-aligned and raw-gradient predictions at early, middle, and late checkpoints, with 16 candidate-update comparisons

per stage. Predictions use the proxy-gradient samples, while observed improvement is evaluated on an independent diagnostic set. Both predictors are compared against the loss change induced by the same candidate optimizer update. Here J is the weighted combination of the answer, visual, and specificity objectives, rather than answer loss alone. For proxy samples (P) and independent diagnostic samples (D), we write

$$J^T(\theta) = \sum_{x \in \{a,v,s\}} w_x J_x^T(\theta), \qquad T \in \{P, D\}. \tag{40}$$

The component objectives follow Equations 10–12 on their respective samples. The same weights w_x are used for both predictors and the diagnostic observation, and are held fixed when comparing the objective before and after a candidate update. This scalar diagnostic objective is distinct from the standardized, EMA-smoothed regional score used for allocation. The optimizer-aligned prediction and observation are

$$\widehat{\Delta J}_b^P = \langle u_b, \nabla_\theta J^P(\theta) \rangle, \qquad \Delta J_b^D = J^D(\theta) - J^D(\theta - u_b). \tag{41}$$

Candidate updates at a checkpoint start from the same model and optimizer states and are evaluated separately. Positive observed values indicate a decrease in the diagnostic objective. The diagonal in Figure 4a is an ideal-agreement reference; the fitted lines summarize the measured prediction–observation pairs.

Optimizer-aligned prediction has higher pooled Pearson correlation than raw-gradient prediction. This pooled comparison summarizes the sampled checkpoints. These are descriptive statistics over candidate-update comparisons, not independent training seeds or an A/V/S regional-rank correlation analysis. They support the utility estimator at the measured checkpoints; the training-level comparison in Appendix F.5 addresses its effect on final performance.

Figure 4b reports answer NLL on diagnostic data independent of the samples used for proxy gradients. We show the V and S diagnostic groups separately, pairing correct images with no images for V and with matched incorrect images for S. From early to late training, loss decreases under all three input conditions, with a larger reduction for correct images in both groups. Larger counterfactual-minus-correct-image gaps therefore accompany better correct-image answer loss on these independent sets. This supports the intended behavior of the allocation proxies; it does not by itself establish higher diagnostic accuracy or improved visual grounding.

Figure 4c reproduces the Vision-Flan-186K results from Table 4. Adding V to A and then S to A+V improves aggregate benchmark performance with the evidence regions and optimizer-aligned allocation mechanism held fixed. These are conditional component comparisons; they do not isolate S alone or establish V/S synergy. Together, the diagnostics and ablation support the measured utility estimates and the benefit of counterfactual objectives within this framework. The benchmark panel visualizes the existing ablation, rather than an additional experiment or diagnostic-set accuracy measurement. The panels report point estimates without uncertainty intervals.

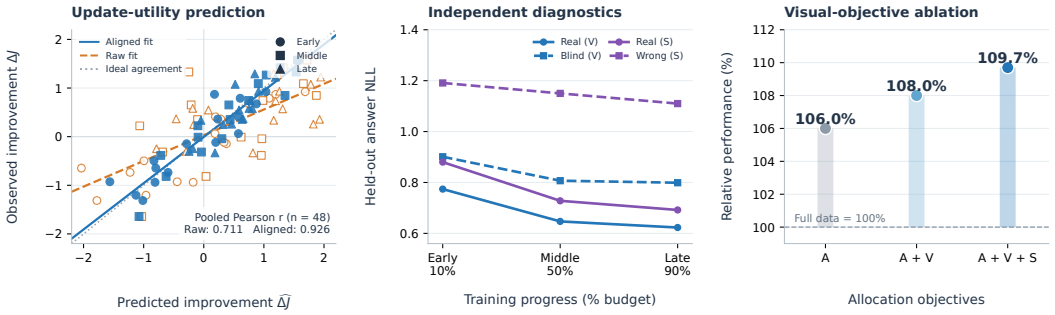
F.4 Average Mixture versus State-Dependent Allocation

Table 9 compares initialized and retrospectively averaged fixed mixtures with state-dependent allocation. DOVAS-Static holds the initialized distribution π_0 fixed. The *Average-mixture static* control instead uses the realized regional training shares from a completed DOVAS run. If b_k is the region used for training update k in that run, define

$$n_b = \sum_{k=1}^{K} \mathbf{1}\{b_k = b\}, \qquad \bar{\pi}_b = \frac{n_b}{\sum_{j=0}^{M} n_j}. \tag{42}$$

Here K counts completed parameter updates, including the initial training block; selection probes are excluded and repeated visits are counted. Because Algorithm 1 draws one region per training update, $\bar{\pi}$ measures the realized share of regional updates, rather than an unweighted average of the probabilities at refreshes or a token/FLOP share.

The control restarts from the same model initialization and initial optimizer state, samples regions from the fixed $\bar{\pi}$ throughout training, and retains the same evidence partition, within-region sampling, batch formation, training optimizer, and learning-rate schedule. It does not replay the original



(a) Update-utility prediction. (b) Independent visual diagnostics. (c) Visual-objective ablation.

Figure 4: **Update-utility validation and visual diagnostics.** (a) Predicted and observed improvement in the weighted objective J for 16 candidate updates at each of three checkpoints. Both predictors are evaluated against the same optimizer-step effect on independent diagnostic data; lines show linear fits and ideal agreement, and r is pooled Pearson correlation. (b) Independent diagnostic answer NLL under correct-image and counterfactual conditions. Colors distinguish V/S diagnostic groups; solid and dashed curves denote correct and counterfactual inputs. (c) Reported A/V/S ablation on Vision-Flan-186K with LLaVA-v1.5-7B, normalized to the corresponding full-data baseline.

Table 9: **Initial, average, and state-dependent region mixtures.** LLaVA-v1.5-7B with full-parameter fine-tuning. Each evaluated training run uses the default compute cap; Average-mixture static additionally uses a prior DOVAS run as described in Appendix F.4. DOVAS-Static and DOVAS values are reproduced from Tables 1 and 2. Rel. uses the corresponding dataset's Full Data baseline and reports the mean benchmark-wise ratio as a percentage.

Allocation	VQAv2	GQA	LLaVA-W	SQA-I	TextVQA	MME-P	MMB-EN	MMB-CN	POPE	MM-Vet	Rel.
LLaVA-665K											
DOVAS-Static (initial mixture)	76.9	62.4	65.9	66.1	57.9	1497.6	63.1	56.6	84.0	31.3	99.1%
Average-mixture static	76.7	61.8	67.3	65.5	58.3	1501.7	63.5	57.9	86.1	30.6	99.5%
DOVAS (state-dependent)	78.9	63.0	69.2	65.8	59.4	1511.1	65.2	59.3	86.7	30.7	101.1%
Vision-Flan-186K											
DOVAS-Static (initial mixture)	74.2	51.5	42.8	60.9	43.5	1356.1	57.9	55.1	91.5	30.6	106.9%
Average-mixture static	74.4	51.4	43.7	61.3	44.2	1342.9	59.7	55.9	92.4	30.5	107.8%
DOVAS (state-dependent)	76.5	52.0	44.5	61.7	45.1	1378.1	61.1	55.7	94.0	31.4	109.7%

region sequence. This preserves the source run's aggregate regional shares in expectation while removing their dependence on the current state. DOVAS uses the state-dependent distribution $\pi^{(q)}$.

Average-mixture static is a retrospective control: $\bar{\pi}$ is constructed from a completed dynamic run on the same dataset and target model. Its retraining run uses the default online compute cap defined in Appendix D. The prior dynamic run supplies the retrospective mixture and is separate from the evaluated retraining run; its cost is outside the retraining cap, as is offline annotation. This control therefore has access to prior-run information. The fixed mixture avoids dynamic refresh costs during retraining, leaving more of the retraining budget for parameter updates. This is a comparison of allocation policies under that cap, with prior-run information available to the average-mixture control.

On both datasets, Average-mixture static outperforms the initialized fixed mixture in aggregate, while DOVAS achieves the strongest overall performance. Thus, the source run's aggregate allocation provides a useful fixed mixture, but does not recover the benefit of adapting allocation to the current state. DOVAS improves over Average-mixture static across the LLaVA benchmarks and most Vision-Flan benchmarks, with MMB-CN as the exception. These results support adapting which regions are sampled over training beyond choosing their overall mixture. The comparison matches compute budgets rather than update counts, so the evidence concerns the complete allocation policy under the cap rather than sample ordering in isolation.

Table 10: **Raw-gradient versus optimizer-aligned dynamic allocation.** LLaVA-v1.5-7B with full-parameter fine-tuning at the default online compute cap. Both variants use all A/V/S objectives and the same allocation rule; the regional update vector in the utility estimator changes. DOVAS values are reproduced from Tables 1 and 2. Rel. uses the corresponding dataset's Full Data baseline and reports the mean benchmark-wise ratio as a percentage.

Utility estimator	VQAv2	GQA	LLaVA-W	SQA-I	TextVQA	MME-P	MMB-EN	MMB-CN	POPE	MM-Vet	Rel.
LLaVA-665K											
Raw-gradient dynamic	77.4	61.3	67.4	64.6	59.5	1504.9	64.9	58.7	85.5	30.1	99.7%
DOVAS (optimizer-aligned)	78.9	63.0	69.2	65.8	59.4	1511.1	65.2	59.3	86.7	30.7	101.1%
Vision-Flan-186K											
Raw-gradient dynamic	75.4	51.1	42.3	61.8	44.7	1358.3	60.4	54.9	92.0	30.6	107.8%
DOVAS (optimizer-aligned)	76.5	52.0	44.5	61.7	45.1	1378.1	61.1	55.7	94.0	31.4	109.7%

F.5 Contribution of Optimizer-Aligned Utility Estimation

Table 10 isolates the update vector used to score each region. Both variants retain the same fixed evidence regions, A/V/S proxy pools and objectives, regional probe construction, initial distribution, refresh interval, cross-region standardization, EMA smoothing, and softmax temperature. The *Raw-gradient dynamic* control replaces the virtual optimizer update $u_b^{(q)}$ in Equation 16 with the regional training gradient $g_b^{(q)}$:

$$A_{b,\mathrm{raw}}^{(q)} = \langle g_b^{(q)}, g_a^{(q)} \rangle, \quad V_{b,\mathrm{raw}}^{(q)} = \langle g_b^{(q)}, g_v^{(q)} \rangle, \quad S_{b,\mathrm{raw}}^{(q)} = \langle g_b^{(q)}, g_s^{(q)} \rangle. \tag{43}$$

These scores undergo the same calibration and allocation steps as DOVAS. Parameter training uses the same optimizer in both variants; only the utility estimator changes. This contrast tests the complete optimizer transformation used by DOVAS, including its state and parameter-group scaling, rather than attributing any difference to optimizer moments alone. It complements the A/V/S ablation in Table 4, which varies the objectives while retaining optimizer-aligned estimation.

Both variants use the same initialization and the default online compute cap, charging their selection costs under Appendix D. The raw-gradient control omits the virtual-update arithmetic from selection. The comparison therefore matches online compute budgets rather than the number of parameter updates.

DOVAS outperforms Raw-gradient dynamic in aggregate on both datasets, with gains across most benchmarks. TextVQA on LLaVA-665K and SQA-I on Vision-Flan-186K are the exceptions, where the raw-gradient control performs slightly better. Together with the update-level prediction results in Appendix F.3, this controlled ablation supports the benefit of using the optimizer-transformed update to guide allocation beyond raw-gradient scoring.

F.6 Repeatability across Training Seeds

Table 11 reports repeated runs of Random, DOVAS-Static, and DOVAS on both training datasets. The protocol varies the training seed while keeping the model initialization, offline evidence partition, proxy pools, and training configuration fixed. The same seed set is used across methods, and each run follows the default online compute cap in Appendix D. This comparison measures variability due to stochastic training and sampling, conditional on the fixed offline preparation. Each run's Rel. is normalized by the same dataset-specific Full Data reference used in the main tables; the summary reports the mean and sample standard deviation across runs.

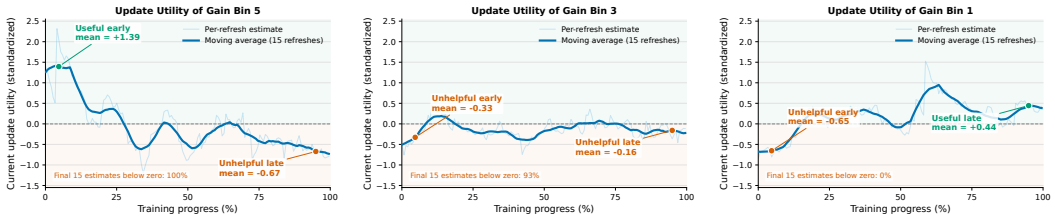
DOVAS achieves the highest mean performance on both datasets and outperforms both baselines for every matched seed. DOVAS-Static also consistently outperforms Random. This ordering is preserved across the observed training variation, supporting the repeatability of the relative performance gains under the fixed offline preparation.

G Qualitative Analysis

We visualize the optimizer-aligned utility trajectories of three representative evidence regions in Figure 5. The regions are constructed before target-model training and remain fixed, so the changes

Table 11: **Repeatability across training seeds.** LLaVA-v1.5-7B with full-parameter fine-tuning under the default online compute cap. Entries report Rel. (%); each dataset block contains individual runs and their mean ± sample standard deviation. Run indices correspond to matched training seeds across methods. Summary statistics are calculated from the listed runs and rounded to two decimal places.

	LLaVA-665K						Vision-Flan-186K					
Method	Run 1	Run 2	Run 3	Run 4	Run 5	Mean ± SD	Run 1	Run 2	Run 3	Run 4	Run 5	Mean ± SD
Random	95.6	95.8	95.9	96.2	96.1	95.92 ± 0.24	96.4	96.4	96.1	96.2	96.9	96.40 ± 0.31
DOVAS-Static	99.0	99.2	99.4	98.9	99.0	99.10 ± 0.20	106.6	107.4	106.8	106.8	106.9	106.90 ± 0.30
DOVAS	101.3	100.8	101.3	101.1	101.0	101.10 ± 0.21	109.7	109.7	109.4	110.1	109.6	109.70 ± 0.25



(a) High-evidence region (Bin 5). (b) Intermediate-evidence region (Bin 3). (c) Low-evidence region (Bin 1).

Figure 5: Evolution of optimizer-aligned update utility for representative evidence regions during training. The light curve shows the utility estimated at each refresh, and the dark curve shows its moving average. Utilities are standardized across regions; negative values indicate below-reference relative utility, not necessarily harmful updates. The trajectories show changing relative allocation preferences while region membership remains fixed.

in their utilities reflect the evolving model and optimizer states rather than a change in region membership.

The high-evidence region (Bin 5) has higher relative utility early in training and lower relative utility later. The low-evidence region (Bin 1) shows the opposite tendency, while the intermediate-evidence region (Bin 3) varies less across the observed trajectory. These observations motivate reassessing the allocation of fixed regions as training progresses.

The trajectories do not establish that a region has been fully learned or that later updates teach more specialized capabilities. They describe relative preferences under the measured proxy objectives, rather than an absolute ranking of data quality or a causal account of capability acquisition.

H Representative Samples across Evidence Regions

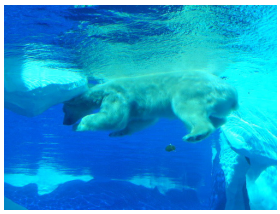
Figures 6– 8 present representative training examples from the three regions analyzed above. Each image is paired with its instruction and reference response. The examples illustrate how the fixed offline partition separates samples with different levels of visual evidence while retaining diverse tasks, including recognition, spatial reasoning, counting, attributes, and text understanding.



B5-1
Instruction. Provide a one-sentence caption for the provided image. Reference OCR token: ALPH, AMERICAN.
Reference response. A silver plane that says American is on display in a museum.



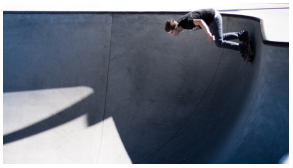
B5-2
Instruction. How many people are in the image?
Reference response. There are four people in the image—a woman standing next to three other people.



B5-3
Instruction. What kind of animal is it?
Reference response. Polar bear.



B5-4
Instruction. What is the man holding in his right hand?
Reference response. Frisbee.



B5-5
Instruction. Is this person wearing any safety gear?
Reference response. No.

Figure 6: Representative samples from the high-evidence region (Bin 5). The corresponding region has above-reference standardized utility early in training; the trajectory does not measure each example's utility.



B3-1
Instruction. What color is the plate the pizza is on?
Reference response. White.



B3-2
Instruction. Provide a one-sentence caption for the provided image.
Reference response. A submarine with a shark face painted on it is docked next to a recycling bin and two trash cans.



B3-3
Instruction. Why are they cooking on a boat? A. feed fish; B. no room; C. tastes better; D. it is home.
Reference response. D.

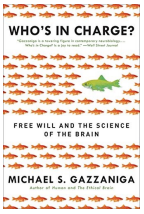


B3-4
Instruction. How many hotdogs are on the plates held by the man?
Reference response. There are five hotdogs on the plates held by the man.



B3-5
Instruction. What is the person to the left of the fence wearing?
Reference response. Helmet.

Figure 7: Representative samples from the intermediate-evidence region (Bin 3). The corresponding region has relatively stable standardized utility over the observed training trajectory.



B1-1
Instruction. Who wrote this book?
Reference response. Michael S. Gazzaniga.



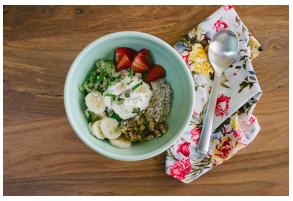
B1-2
Instruction. Provide a one-sentence caption for the provided image. Reference OCR token: Technology, computer, 1980.
Reference response. An old Apple II computer from the 1980s with a purple background that says Technology on it.



B1-3
Instruction. What color are the leaves?
Reference response. Yellow.



B1-4
Instruction. Is that a truck or a bus?
Reference response. Bus.



B1-5
Instruction. What utensil is to the right of the bowl?
Reference response. Spoon.

Figure 8: Representative samples from the low-evidence region (Bin 1). The corresponding region's standardized utility increases during training; these examples illustrate its contents without establishing which capabilities cause that change.