Data Selection for Fine-tuning Vision Language Models via Cross-Modal Alignment Trajectories

Nilay Naharas*[1] Dang Nguyen*[1] Neslihan Bulut[2]
Mohammadhossein Bateni[2] Vahab Mirrokni[2] Baharan Mirzasoleiman[1,2]

[1]University of California, Los Angeles [2]Google Research

ICML 2026

*Equal contribution
Project page: `bigml-cs-ucla.github.io/XMAS-project-page`

The Problem: Most Training Data Is Redundant

Vision-language models learn to look at an image and answer in words. They are fine-tuned on *huge* instruction datasets.

But much of that data is redundant:

- Many examples teach the model the *same thing*
- Training on all of it burns compute, time, and energy
- . . . without making the model any smarter

Two popular datasets

Dataset	Size	XMAS cuts
LLaVA-665k	665K	**−50%**
Vision-Flan	186K	**−85%**

Key Question

Can we **throw away the redundant data** *without* hurting the model?

The Surprise: Nothing Beats Picking at Random

For text-only language models, good data-selection methods exist.

But for **vision-language models**, it turned out to be hard:

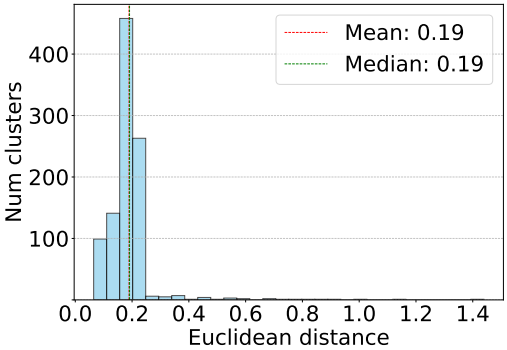
- **No prior method** reliably beats just keeping examples *at random*
- The signals they rely on miss the real target

Why? The model has *billions* of parameters — the principled signal (full gradients) is far too expensive to compute.

Old signal	**Why it misses**
Per-example scores	Rank items, but don't drop *groups* with the same effect
Embedding / dedup	Find look-alikes, not *training-time* redundancy
Full gradients	Principled, but far too costly at billions of params

Our Insight: Watch How the Model "Aligns" Image and Text

- **What is "redundant"?**
  Two examples are redundant if they push the model the *same way* — they produce **similar gradients**.
- **Gradients are too expensive.**
  So we use a cheap stand-in: the **cross-modal attention** — how strongly the *words* attend to the *image* — from a small **proxy model**.
- **Why it works.**
  The proxy's attention closely tracks the big target model's gradients (small L_2 gap, see chart).



A *small* proxy's alignment closely matches the *large* target's behavior ($L_2 < 0.25$).

XMAS in Four Steps

Step 1 — Train a small proxy

Fine-tune a small VLM; save a few checkpoints along the way.

Step 2 — Track each example

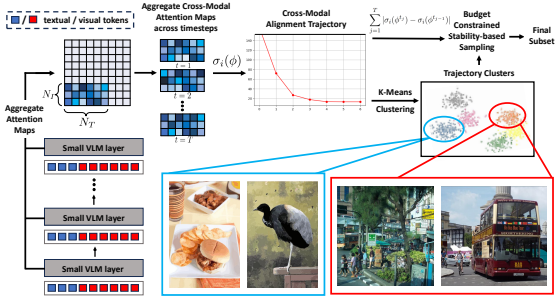
Record how its image–text alignment changes across checkpoints — its *alignment trajectory*.

Step 3 — Group similar behavior

Cluster examples whose trajectories look alike: each cluster is one "type" of training signal.

Step 4 — Keep a balanced, stable subset

From every cluster keep the most *stable* examples — a small, representative sample.

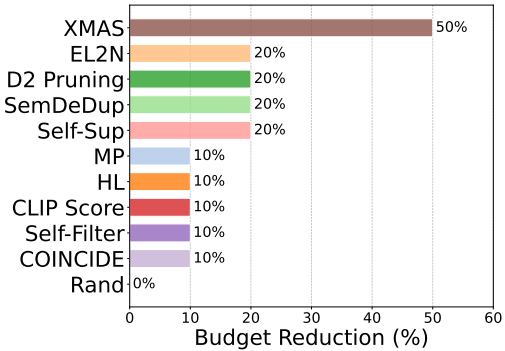


XMAS pipeline: proxy → trajectories → clusters → stable representatives.

Why It Works (in Plain Words)

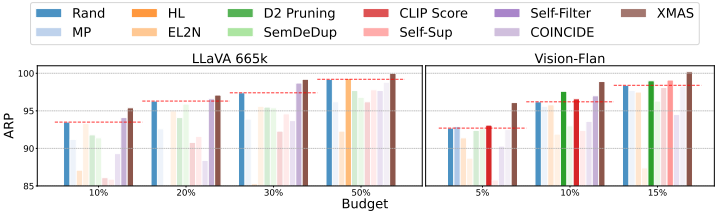
Three guarantees back the method up:

1. **Similar attention ⇒ similar gradients.**
   So examples with look-alike attention really are redundant.
2. **A few checkpoints are enough.**
   They summarize the whole fine-tuning path — no need to track every step.
3. **The subset converges to full-data quality.**
   Balanced cluster representatives land near the full-data solution.



XMAS reaches full-data performance on LLaVA-665k with only **50%** of the data — 30% more reduction than the best baseline.

Results: Less Data, Same Performance, Faster



XMAS (dark bars) is the *only* method that consistently beats random across budgets on both datasets.

- Discard **50%** of LLaVA-665k & **85%** of Vision-Flan
- Keep **100%** performance on **10** benchmarks
- **30%** more reduction than the best baseline
- **1.2**× faster end-to-end (selection included)
- Generalizes across many target & proxy models

Take-Away

One idea

Be **smart about which data we keep**: cluster examples by how a small proxy *aligns* image and text, then keep a balanced, stable sample.

XMAS in a nutshell:

1. Train a small proxy and track *alignment trajectories*
2. Cluster redundant examples; keep the most stable representatives
3. Same performance, far less data

Why it matters:

- Powerful multimodal AI that is **cheaper, faster, more sustainable** to train
- A guide for *how to collect* better data in the future

Project page: `bigml-cs-ucla.github.io/XMAS-project-page` — Thank you!