Flamingo → InternVL3.5

From frozen backbones to large VLMs

Seminal anchor: Flamingo — Alayrac et al., NeurIPS 2022
Frontier anchor: InternVL3.5 — Wang et al., arXiv 2025

Kun-lin Hsieh, Pratham Sunkad, Yiran Luo
CS 8803-VLM · Sep 8th

Georgia Tech

Outline

- **Problem statement & related work**
- **Flamingo**: architecture, data, and few-shot learning
- **InternVL3.5**: architecture, data, pretraining, and SFT
- **Post-training & efficiency**: Cascade RL, ViCO, deployment, and TTS
- **Experiments & results**: selected tasks and benchmarks
- **What persisted and what changed?**
- **Limitations, societal implications & discussion**

Running question

What stays essential as a frozen-backbone interface becomes a large, post-trained VLM?

From frozen backbones to large VLMs

Flamingo (2022) → InternVL3.5 (2025)

How should pretrained vision and language become one reusable system?

01

Build the interface

Flamingo: methods, data, and few-shot evidence.

02

Train the full system

InternVL3.5: scale, alignment, and efficiency.

03

Compare the choices

What persists, what changes, and what to test next.

Same goal: a reusable vision–language model. Different answers to where adaptation belongs.

Alayrac et al. (2022), Flamingo, §1–2; Wang et al. (2025), InternVL3.5, §1–2

Related work: reuse, connect, then instruct

2021 — **Reusable components**

CLIP aligns images and text. Frozen explores visual prompting of a frozen LM.

2022 — **Flamingo: a multimodal interface**

Frozen vision + frozen LM + learned visual memory and cross-attention.

2023 → — **Several paths forward**

BLIP-2: query-based connection. LLaVA: visual instruction tuning.

These are neighboring directions—not a claim of a single linear ancestry.

Frozen, 2021; BLIP-2, 2023; Visual Instruction Tuning, 2023; Flamingo, §1 and §4

What persisted—and what changed in the interface?

Flamingo

64 visual tokens → Cross-attention

Separate visual memory
Frozen LM layers

Fixed 64 per image/video; not per frame.

InternVL3.5

ViT + MLP → Visual + text tokens

Visual tokens inside the LM
Joint multimodal pretraining

Flash: 64 or 256 per high-resolution tile.

Both translate visual evidence into language; they expose that evidence differently.

Alayrac et al. (2022), Flamingo, §2.1–2.3; Wang et al. (2025), InternVL3.5, §2.1

Flamingo: a Visual Language Model for Few-Shot Learning

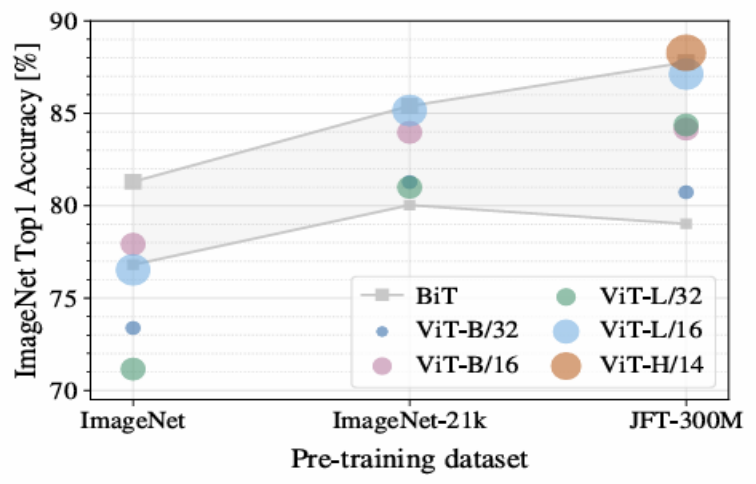
Georgia Tech

Predecessors: ViT and CLIP for Classification

ViT

Key Takeaways:

1) Large Scale Training trumps inductive bias for vision-transformers
2) Improvement in Pre-training compute efficiency and achieved SOTA on classification post finetuning

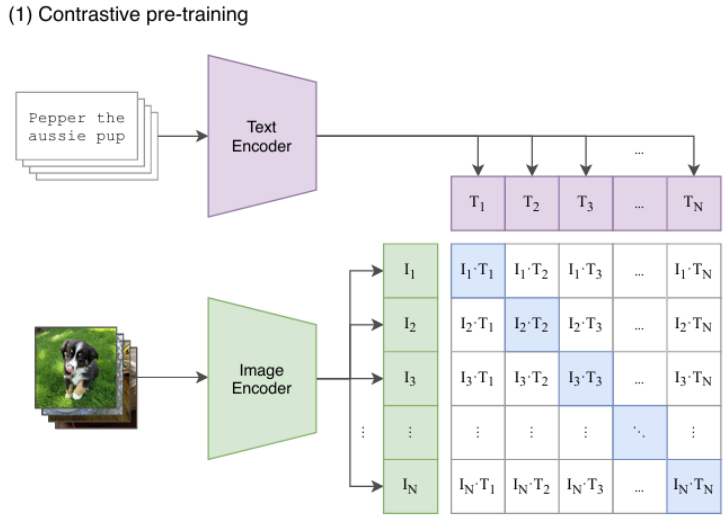


Limitation: Required labelled dataset; only performed classification; Lagging performance in SSL

CLIP

Key Takeaways:

1) Contrastive Learning leading to robust zero-shot transfer due to rich visual features learnt by the encoder
2) Use image-text pair data scrapped through the web



Limitation: Not generative in nature; Cannot handle interleaved combination of image/video and text data

Flamingo: The Main Idea

1) In-context Few Shot Learning :

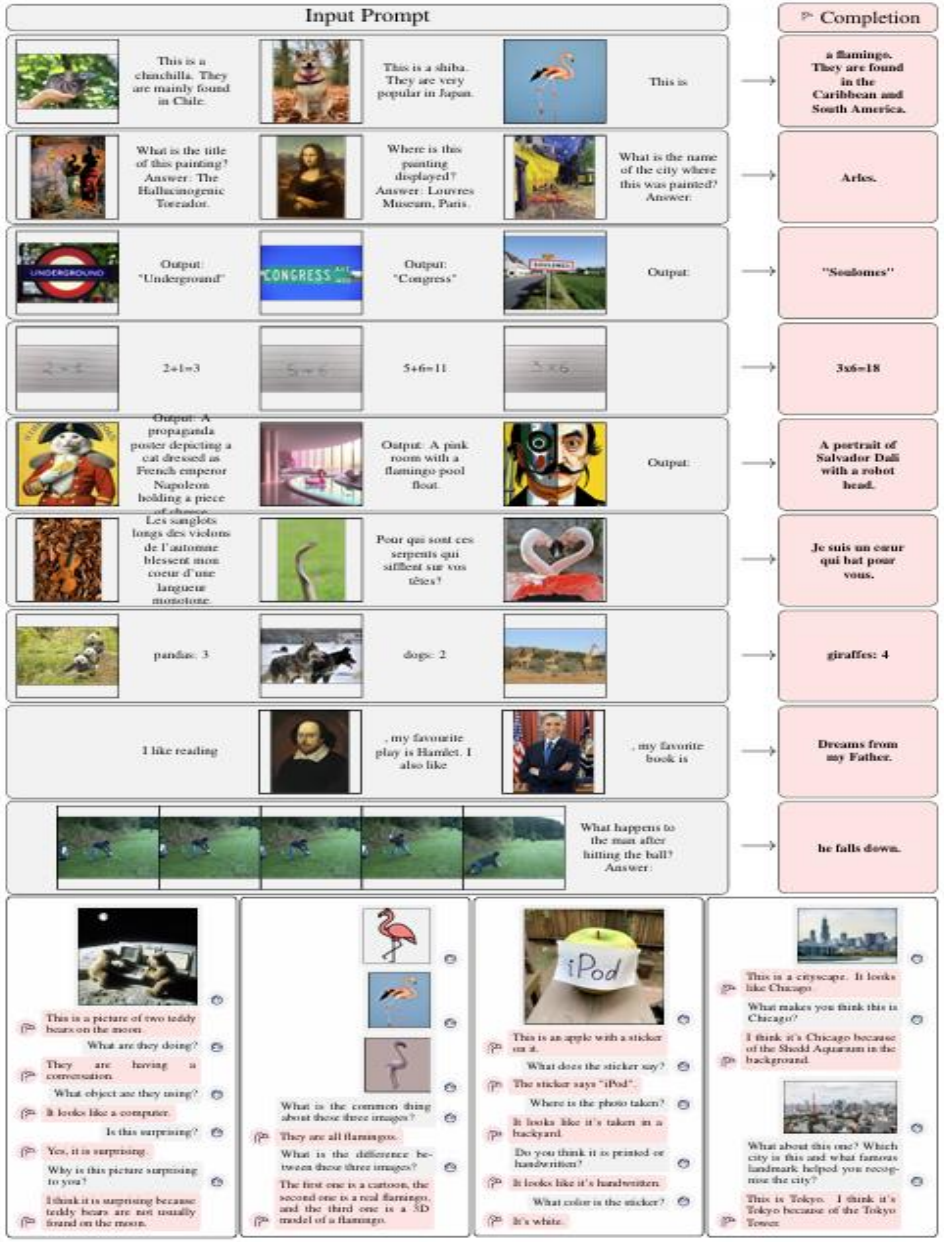
Can a multi-modal model adapt to novel tasks using a handful of annotated examples?

2) Handle sequences of arbitrary interleaved text and images/videos

3) Bridge powerful pretrained vision only and language only models

Inspiration: Large Language Models
LLMs are good few shot learners – adapt to the prompt

Challenge: Do this for image/video with text generation
Solution: Text prediction problem with visual input conditioning



Approach

Flamingo models the likelihood of text y conditioned on interleaved images and videos x as follows:

$$p(y|x) = \prod_{\ell=1}^{L} p(y_\ell | y_{<\ell}, x_{\leq \ell}),$$

y_ℓ - The ℓ-th language token in the output text

$y_{<\ell}$ - all text tokens generated before position ℓ

$x_{\leq \ell}$ - all images/videos in the interleaved sequence that appeared before token y_ℓ

L - total number of text tokens

p - parameterized by the Flamingo model itself

Paper Contributions

Introduce the Flamingo family of VLMs which can perform various multimodal tasks (such as captioning, visual dialogue, or visual question-answering) from only a few input/output examples.

Evaluate how Flamingo models can be adapted to various tasks via few-shot learning.

Flamingo sets a new state of the art in few-shot learning on a wide array of 16 multimodal language and image/video understanding tasks. On 6 of these 16 tasks, Flamingo also outperforms the fine-tuned state of the art despite using only 32 task-specific examples

Architecture: preserve the backbones, learn the link

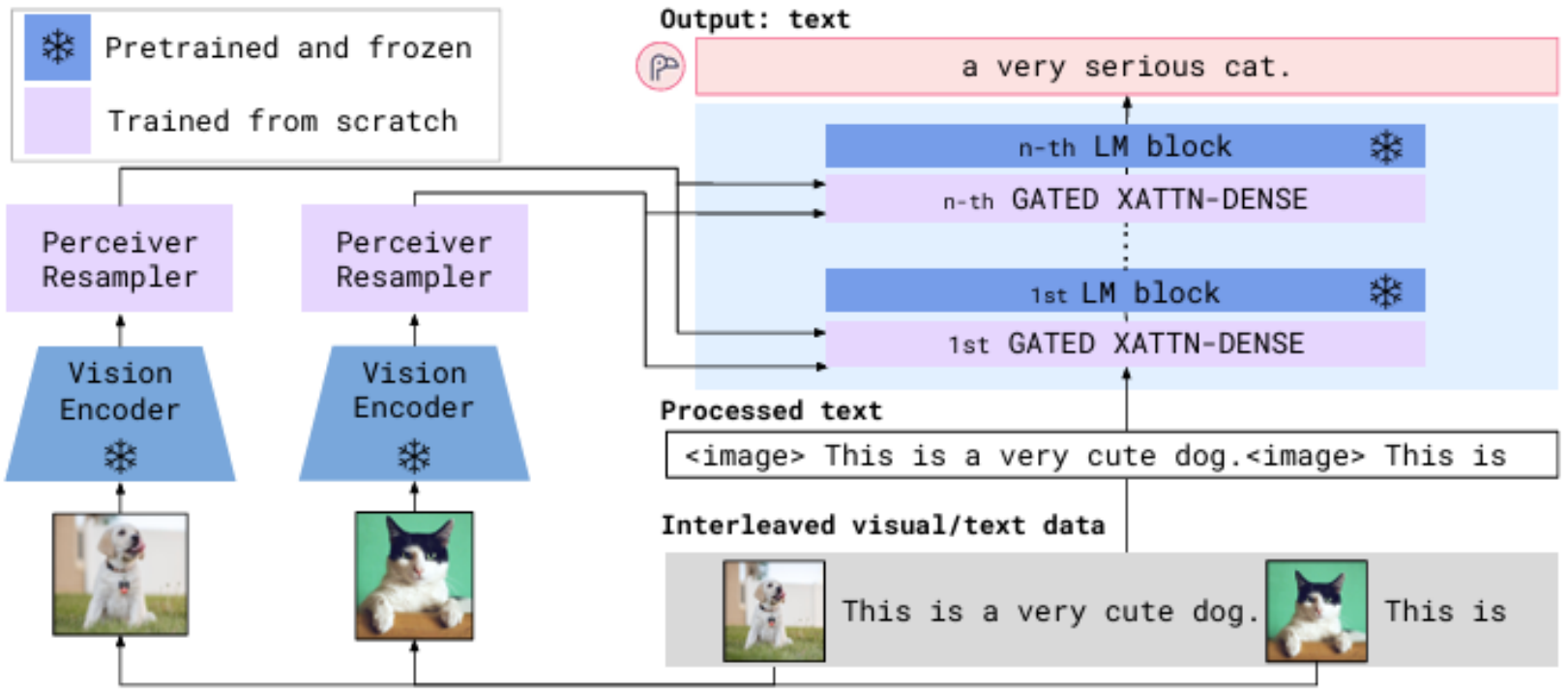


Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

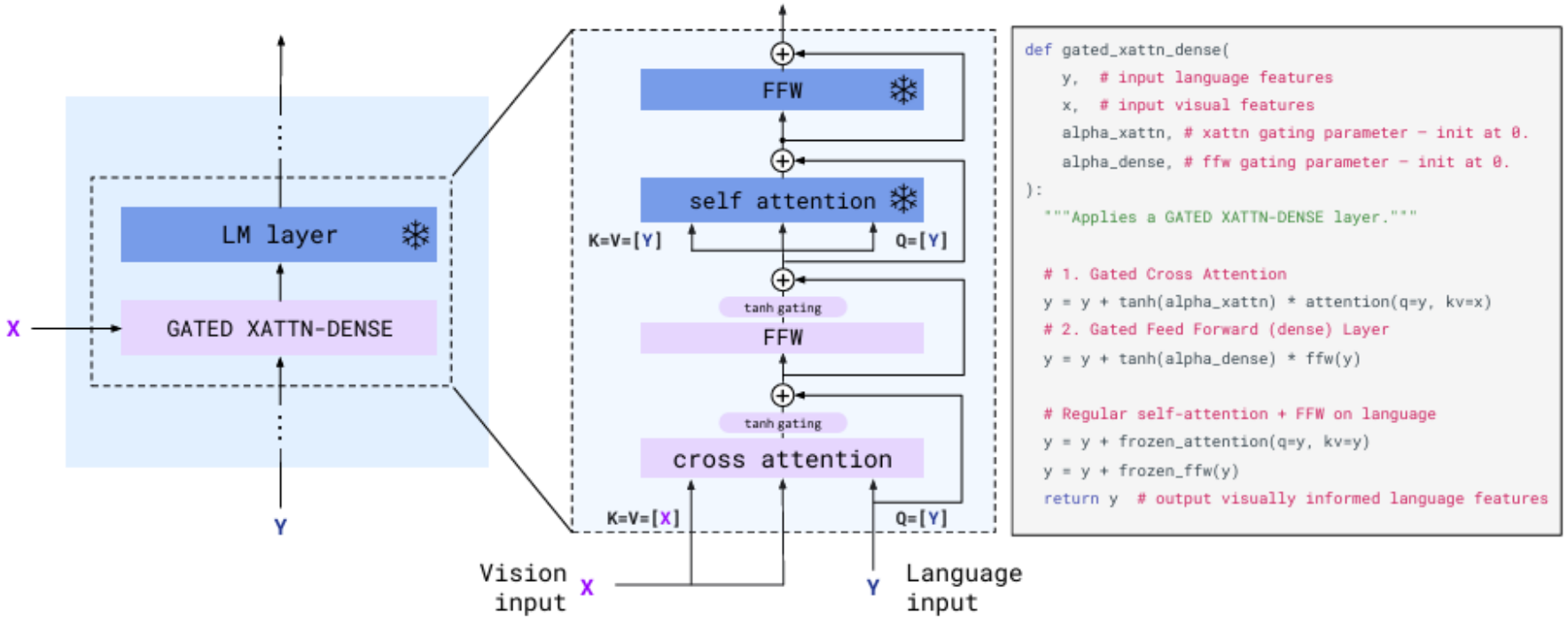
Vision Encoder:
Frozen and PreTrained using Contrastive Loss. NFNet-F6 or CLIP ViT

Text Encoder:
Frozen and Pretrained, Chinchilla Models(1.4B, 7B, 70B)

Perceiver Resampler:
Takes in variable number of image/video features from vision encoder and produces fixed number of visual outputs (64)

Gated XATTN-DENSE:
Cross attention layers between image and text with causal masking

GATED XATTN-DENSE Layers




```
def gated_xattn_dense(
    y,  # input language features
    x,  # input visual features
    alpha_xattn, # xattn gating parameter – init at 0.
    alpha_dense, # ffw gating parameter – init at 0.
):
  """Applies a GATED XATTN-DENSE layer."""

  # 1. Gated Cross Attention
  y = y + tanh(alpha_xattn) * attention(q=y, kv=x)
  # 2. Gated Feed Forward (dense) Layer
  y = y + tanh(alpha_dense) * ffw(y)

  # Regular self-attention + FFW on language
  y = y + frozen_attention(q=y, kv=y)
  y = y + frozen_ffw(y)
  return y  # output visually informed language features
```

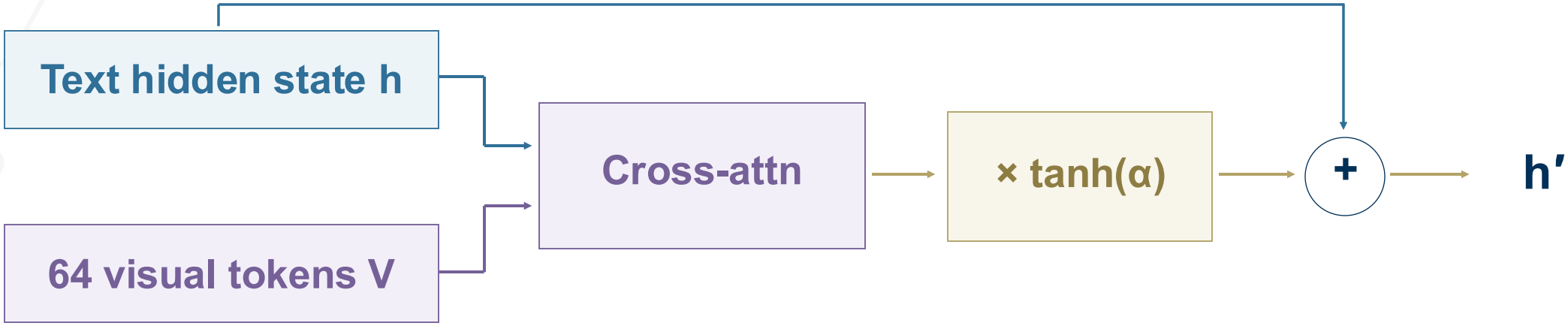
Figure 4: **GATED XATTN-DENSE layers.** To condition the LM on visual inputs, we insert new cross-attention layers between existing pretrained and frozen LM layers. The keys and values in these layers are obtained from the vision features while the queries are derived from the language inputs. They are followed by dense feed-forward layers. These layers are *gated* so that the LM is kept intact at initialization for improved stability and performance.

Alayrac et al. (2022), Flamingo, §2.1; App. A.1.1

Georgia Tech

Gated cross-attention starts as the original LM

TEXT AS QUERIES · VISUAL MEMORY AS KEYS AND VALUES



$$h' = h + \tanh(\alpha)\,\mathrm{CA}(h, V)$$

At α = 0, h′ = h. The following feed-forward residual is gated too.

Zero-initialized gates let the visual pathway enter gradually.

Alayrac et al. (2022), Flamingo, §2.2; Fig. 4

Each text span directly sees the latest image

DIRECT CROSS-ATTENTION · GROUPED BY IMAGE/TEXT SPAN

	Image 1	Image 2	Image 3
Text after image 1	yes	—	—
Text after image 2	—	yes	—
Text after image 3	—	—	yes

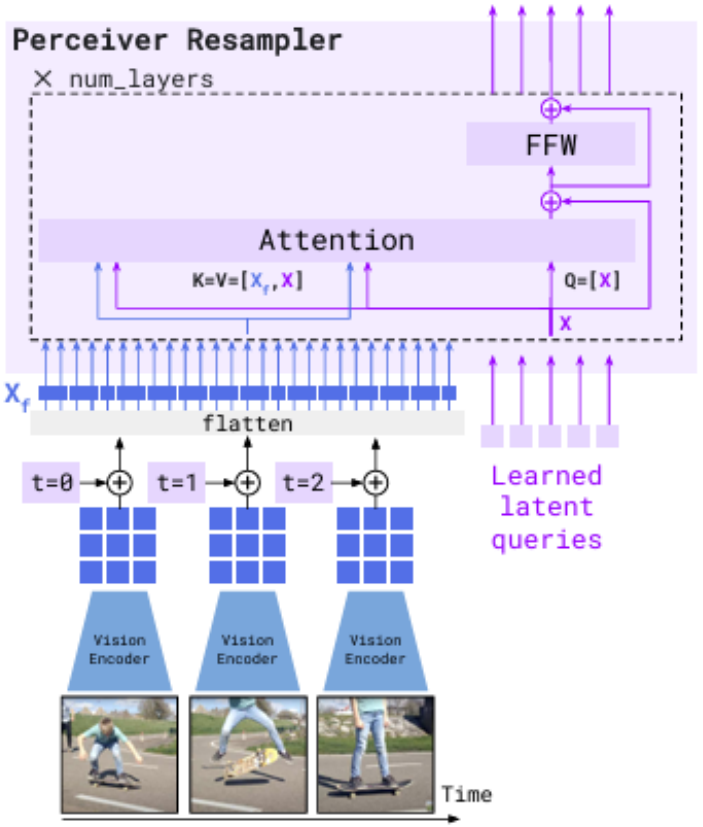
A fixed-size visual memory for each text span.

≤5 images in training sequences; up to 32 image/video examples evaluated at inference.

INDIRECT PATH THROUGH TEXT SELF-ATTENTION

Earlier images remain accessible indirectly through the text hidden states.

Perceiver Resampler




```python
def perceiver_resampler(
    x_f,  # The [T, S, d] visual features (T=time, S=space)
    time_embeddings,  # The [T, 1, d] time pos embeddings.
    x,  # R learned latents of shape [R, d]
    num_layers,  # Number of layers
):
  """The Perceiver Resampler model."""

  # Add the time position embeddings and flatten.
  x_f = x_f + time_embeddings
  x_f = flatten(x_f)  # [T, S, d] -> [T * S, d]
  # Apply the Perceiver Resampler layers.
  for i in range(num_layers):
    # Attention.
    x = x + attention_i(q=x, kv=concat([x_f, x]))
    # Feed forward.
    x = x + ffw_i(x)
  return x
```

Figure 5: **The Perceiver Resampler** module maps a *variable* size grid of spatio-temporal visual features output by the Vision Encoder to a *fixed* number of output tokens (five in the figure), independently from the input image resolution or the number of input video frames. This transformer has a set of learned latent vectors as queries, and the keys and values are a concatenation of the spatio-temporal visual features with the learned latent vectors.

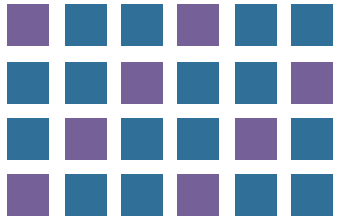
Alayrac et al. (2022), Flamingo, §2.1; App. A.1.1

Georgia Tech

Perceiver Resampler: 64 learned queries

VARIABLE INPUT LENGTH → FIXED VISUAL MEMORY

Visual features X

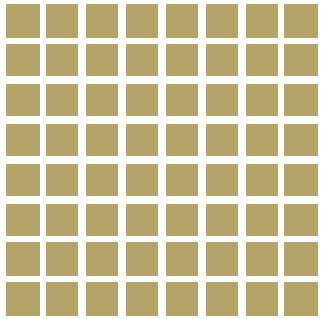


Many spatial or temporal features.

Latent queries Z

Read all features

64 output tokens



8 × 8 drawn here = 64 slots

$$Q = Z, \qquad \mathrm{K}, \mathrm{V} = [X; Z]$$

The output count is fixed; what the queries extract is content-dependent.

Four web datasets teach different things

DATASET SIZES ARE RECORD COUNTS, NOT MIXTURE PERCENTAGES

Dataset	Scale	Illustrative record format	Learning signal
M3W	43M pages	[image] paragraph [image] text	Interleaved context
ALIGN	1.8B pairs	Image + short alt-text	Broad visual concepts
LTIP	312M pairs	Image + longer description	Richer detail
VTP	27M clips	Short video + sentence	Temporal grounding

M3W training window: 256 text tokens + at most 5 images, with <image> and <EOC> markers.

Example formats above are schematic; captions are not benchmark answers.

Paired data teaches visual grounding; interleaved data teaches the prompt format.

Images and videos use the same visual front end

One image

A spatial grid of features is flattened into a sequence.

Frozen CNN; not the original ViT.

A video

1 FPS → shared NFNet per frame → add temporal embeddings → flatten.

More frames create more input features.

Encode space and time first; the resampler then fixes the language-side budget.

Learn the interface with next-token prediction

$$L = \sum_m \lambda_m \, \mathrm{E}\left[-\sum_t \log p_\theta(y_t \mid y_{1:t-1}, x_{\le t}) \right]$$

MODEL FAMILY · ROUNDED PARAMETER COUNTS

Flamingo	Frozen LM	New modules ≈
3B	1.4B	1.4B
9B	7B	1.8B
80B	70B	10B

One weighted mixture

M3W: 1.0
ALIGN / LTIP: 0.2 each
VTP: 0.03

Expectation is over examples from each dataset; only the newly added modules are trained here.

"Frozen" saves retraining the backbones; it does not mean the bridge is tiny.

Evaluation: adapt through examples, not gradients

5

Development tasks

Used for design and hyperparameter decisions.

11

Held-out tasks

Not used to choose the architecture or settings.

0 / 4 / 32

Prompt examples

Main table's reported shot counts.

Open answers vs. supplied choices

Generate text, or score candidate answers.

"Zero-shot" is not an empty prompt

Two text-only examples set the format; the query still includes its image.

VQA: answer accuracy. Captioning: CIDEr. Do not average their raw scores.

Read every score together with its task, shot count, and adaptation protocol.

32-shot transfer is strong—but uneven

FLAMINGO-80B · 32 EXAMPLES · NO TASK-SPECIFIC WEIGHT UPDATE

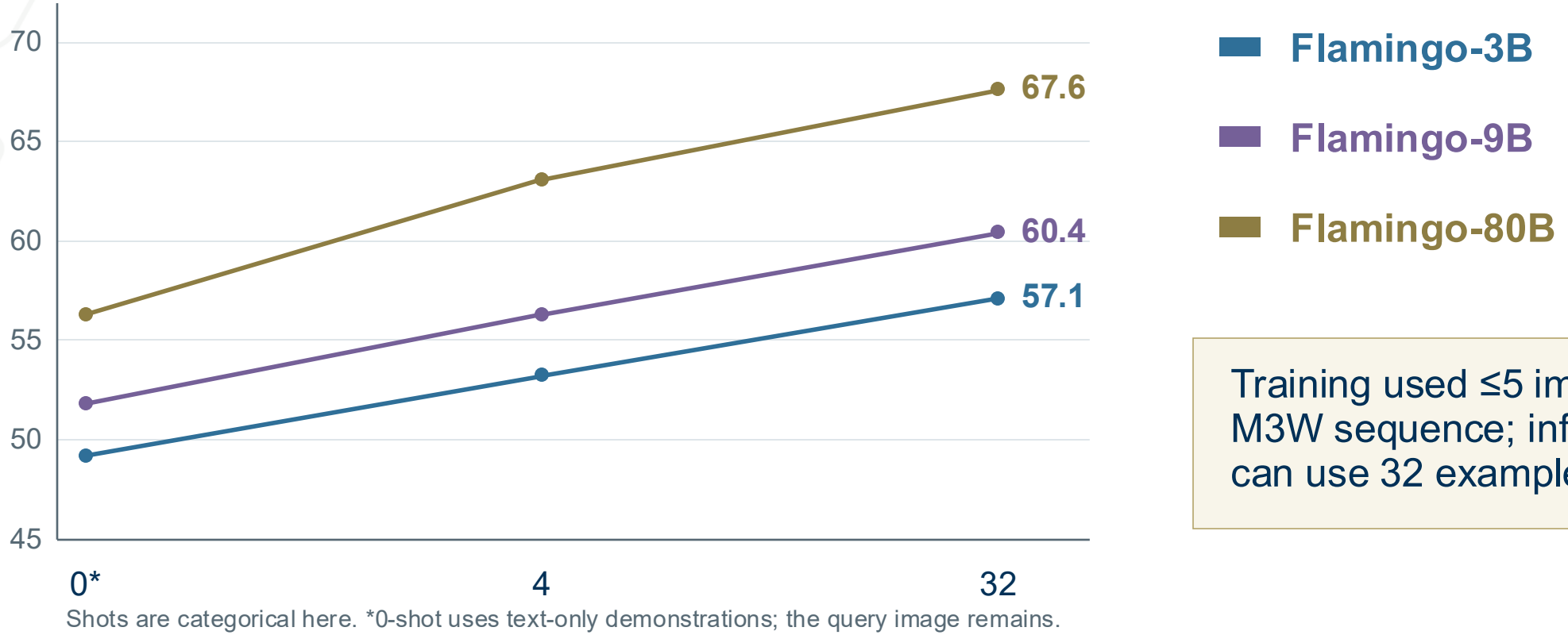
Task	Metric ↑	Flamingo	Fine-tuned baseline
OK-VQA · knowledge QA	**Accuracy**	**57.8**	**54.4**
VQAv2 · image QA	Accuracy	67.6	80.2
MSVD-QA · video QA	**Accuracy**	**52.3**	**47.9**
COCO · image captions	CIDEr	113.8	143.3

Historical baselines from the same table. Different tasks and protocols—not a single accuracy scale.

A few prompt examples can beat some task-specific systems, but not all.

Both model size and prompt examples help

VQAv2 ACCURACY ↑ · TABLE 1 RE-PLOTTED



Shots are categorical here. *0-shot uses text-only demonstrations; the query image remains.

Training used ≤5 images per M3W sequence; inference can use 32 examples.

The same frozen-backbone design benefits from a larger LM and more examples.

Alayrac et al. (2022), Flamingo, Table 1; Fig. 2

Georgia Tech

Ablations explain why the pieces matter

3B MODEL · SHORT TRAINING · 4 SHOTS · FIVE DEVELOPMENT BENCHMARKS

Change	Overall ↑	Δ
Full recipe	**70.7**	—
Without M3W interleaving	53.4	−17.3
Without image–text pairs	60.9	−9.8
Without tanh gates	66.5	−4.2
MLP instead of resampler	66.6	−4.1
Unfreeze pretrained LM	62.7	−8.0
Attend to all earlier images	63.5	−7.2

Overall = average of scores normalized by task-specific SOTA; not raw accuracy.

Largest drop: remove M3W

The training sequence must teach the model how images and text alternate.

Evidence for the recipe, not a universal law.

Interleaving, gating, resampling, and freezing each matter in the tested recipe.

Flamingo also supports supervised fine-tuning

VQAv2 TEST-DEV ACCURACY · TABLE 2

67.6 → **82.0**

32 prompt examples

Task fine-tuning

Different annotation budget and optimization—not a free 32-shot gain.

What changes?

Task-specific training data

Vision encoder is unfrozen

Image resolution: 320 → 480

Base LM remains frozen

Few-shot prompting is the main result—not the only adaptation method tested.

Strengths and limits belong in the same picture

What is convincing

One generative interface

Images, videos, captions, and answers.

Low task-specific data needs

A few examples can specify a useful task.

Held-out benchmark design

11 tasks are excluded from design choices.

What remains limited

Not a low-compute model

Private pretraining + ≈10B added parameters at 80B.

Grounding is not guaranteed

Plausible language can contradict visual evidence.

Prompting has a cost

Sensitive to examples; context and compute grow.

Strong low-data transfer does not imply low-cost training or complete grounding.

InternVL3.5: scale the whole VLM system

A broader, more practical VLM—not just a new backbone.

SEE THE DETAILS

Documents, charts, screenshots, multiple images, and video.

LEARN TO RESPOND

Instructions, reasoning traces, preferences, and rewards.

CONTROL THE COST

Adaptive visual tokens and specialized serving.

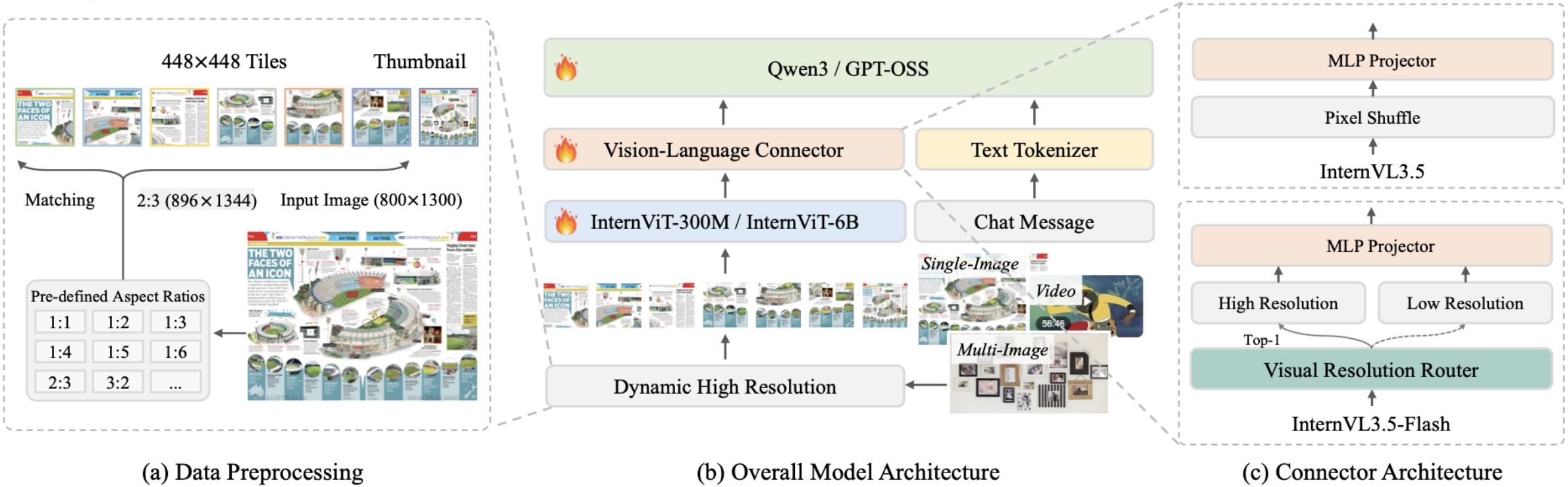
OUR RUNNING QUESTION

What enters the model—and what supervision tells it how to use that information?

A familiar architecture becomes a broad system through its data, objectives, and deployment recipe.

Wang et al. (2025), InternVL3.5, Abstract, §1 and Figures 2–3.

The core architecture: ViT → MLP → LLM



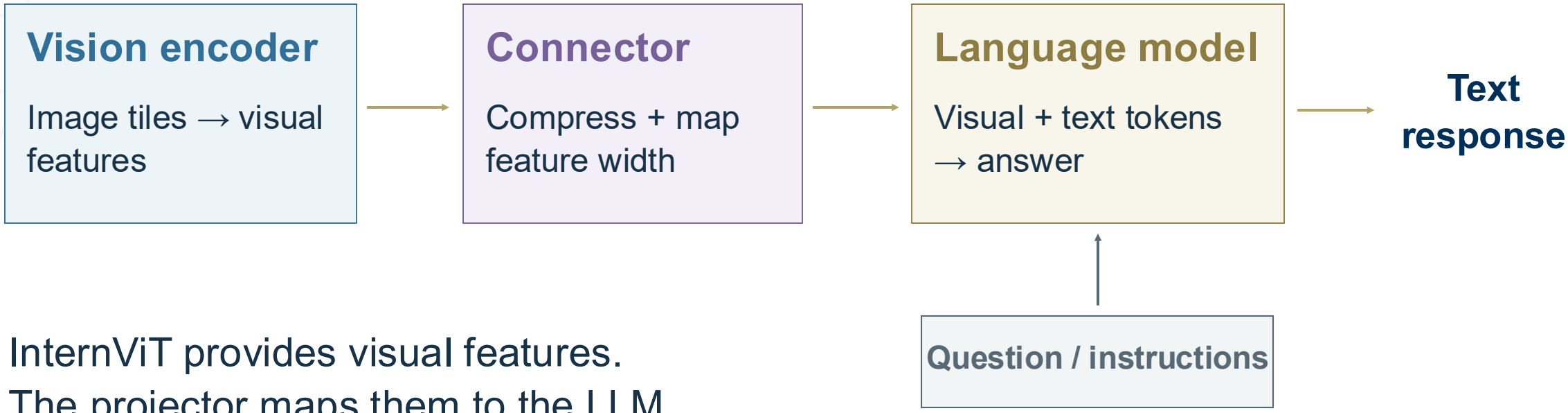
(a) Data Preprocessing (b) Overall Model Architecture (c) Connector Architecture

Figure 2: **Overall architecture.** InternVL3.5 adopts the "ViT–MLP–LLM" paradigm as in previous versions. Building upon InternVL3.5, we further introduce InternVL3.5-Flash, which is extended with an additional visual resolution router (ViR) to dynamically select the appropriate compression rate (*e.g.*, $\frac{1}{4}$ or $\frac{1}{16}$) for each image patch. Unlike Dynamic High Resolution which only splits image patches from the perspective of image width and height, our proposed ViR further introduces adaptivity from the perspective of semantic content.

Wang et al. (2025), InternVL3.5, Abstract, §1 and Figure 2.

The core architecture: ViT → MLP → LLM (Simplified View)

ONE MODEL, TWO KINDS OF INPUT



InternViT provides visual features.
The projector maps them to the LLM.

The connector changes representation; the LLM supplies the shared generative interface.

InternVL3.5, §2.1 and Figure 2. Diagram is a simplified reconstruction.

The core architecture: ViT → MLP → LLM

Model	Vision Encoder	Language Model	#Param		
			Vision	Language	Total
		Dense Models			
InternVL3.5-1B	InternViT-300M	Qwen3-0.6B	0.3B	0.8B	1.1B
InternVL3.5-2B	InternViT-300M	Qwen3-1.7B	0.3B	2.0B	2.3B
InternVL3.5-4B	InternViT-300M	Qwen3-4B	0.3B	4.4B	4.7B
InternVL3.5-8B	InternViT-300M	Qwen3-8B	0.3B	8.2B	8.5B
InternVL3.5-14B	InternViT-300M	Qwen3-14B	0.3B	14.8B	15.1B
InternVL3.5-38B	InternViT-6B	Qwen3-32B	5.5B	32.8B	38.4B
		MoE Models			
InternVL3.5-20B-A4B	InternViT-300M	GPT-OSS-20B	0.3B	20.9B	21.2B (A4B)
InternVL3.5-30B-A3B	InternViT-300M	Qwen3-30B-A3B	0.3B	30.5B	30.8B (A3B)
InternVL3.5-241B-A28B	InternViT-6B	Qwen3-235B-A22B	5.5B	235.1B	240.7B (A28B)

Table 1: **Pre-trained models used in the InternVL3.5 series**. "A" denotes the number of activated parameters.

Note: The vision encoder was initialized with InternViT-300M and InternViT-6B, while the language model was initialized with the Qwen series and GPT-OSS

Wang et al. (2025), InternVL3.5, Abstract, §1 and Table 1.

One recipe, several model scales

ARCHITECTURE › MODEL FAMILY AND MIXTURE OF EXPERTS

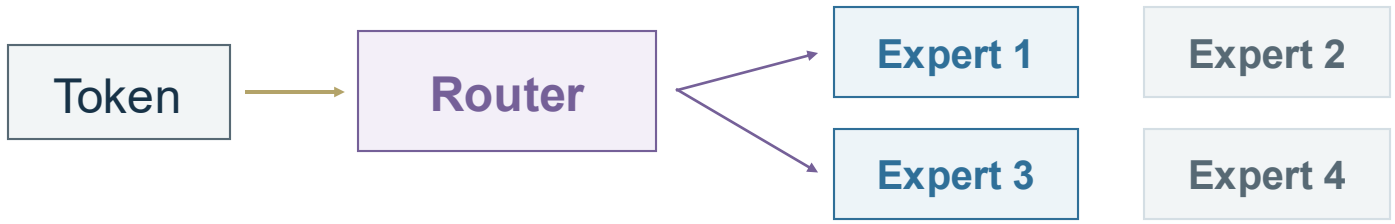
Variant	Vision encoder	Language model
8B	InternViT-300M	Qwen3-8B
38B	InternViT-6B	Qwen3-32B
30B-A3B	InternViT-300M	Qwen3-30B-A3B
241B-A28B	**InternViT-6B**	**Qwen3-235B-A22B**

241B-A28B

241B total parameters
≈28B active per token

MoE routes each token to a subset of experts.

MIXTURE OF EXPERTS



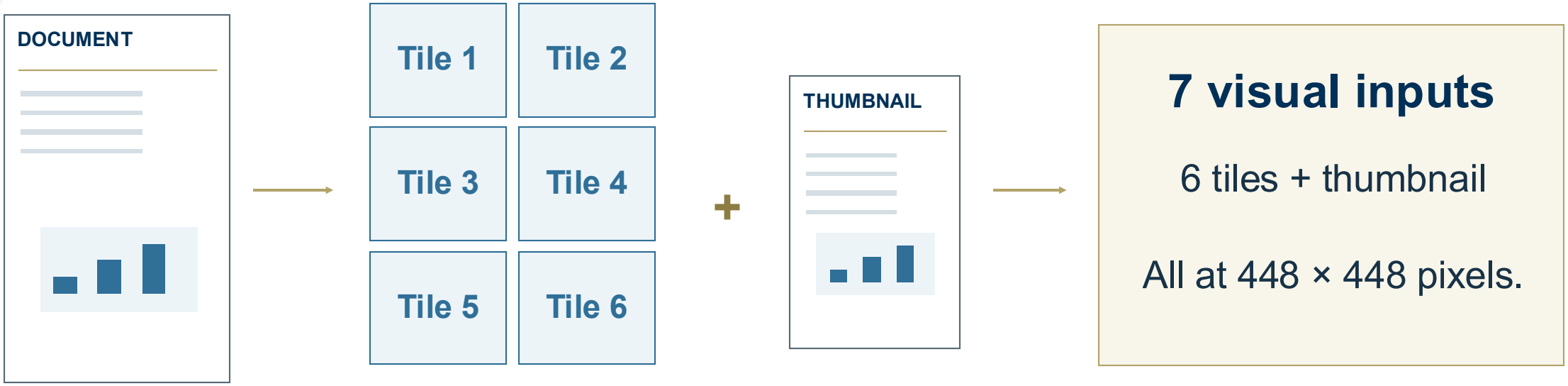
Schematic only:
selected experts shown in blue.

Sparse activation reduces per-token computation; all expert weights still need to be stored.

InternVL3.5, Table 1. Model names are size labels; architecture counts are rounded.

Dynamic high resolution: a map plus close-ups

A single small resize can erase tiny text. Keep local detail and a global view.



Tiling retains readable detail; the thumbnail retains the overall layout.

InternVL3.5, Figure 2 and §2.1; dynamic high resolution inherited from InternVL1.5.

From patch features to language-visible tokens

ARCHITECTURE › PIXEL SHUFFLE AND PROJECTION

ONE 448 × 448 TILE

32 × 32 features

1,024 visual tokens

→

16 × 16 features

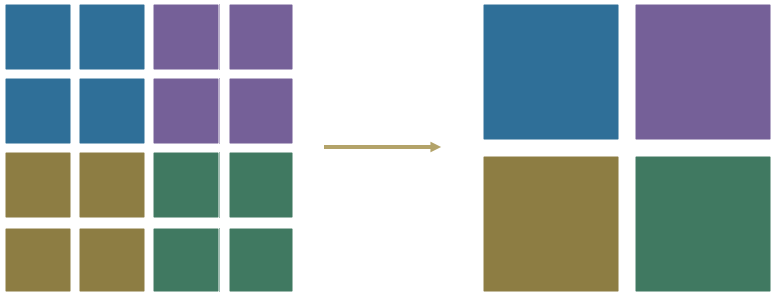
256 grouped tokens

→

MLP projection

Match the LLM width

LOCAL GROUPING: A SMALL SCHEMATIC



Group nearby 2 × 2 features into channels, then project.

The result is still vectors, not caption words.

7 × 256 = 1,792

visual tokens in the preceding example

Compression shortens the LLM sequence while retaining a spatially structured visual representation.

InternVL3.5, Figure 2 and §2.1. Token arithmetic and grouping diagram are explanatory.

Images and text share the language-model context

Visual embeddings and text share the input stream; no caption is inserted first.

One language-model context

ONE IMAGE

Tiles + a thumbnail

MULTIPLE IMAGES

Several visual groups

VIDEO

Ordered frame inputs

The output interface stays text—even when the input is a document, screenshot, or video.

InternVL3.5, §2.1, §2.2 and Figure 2. Sequence diagram is schematic.

Training phases change the learning signal

Pretraining

Broad text + multimodal examples; for vision-language alignment

→

SFT

Selected instructions and reasoning traces; for adaptation to downstream tasks

→

Cascade RL

Preferences + new scored attempts; for improving reasoning performance

→

ViCO (Flash)

Full-budget model as a teacher; minimize output divergence of different visual compression rates

WHAT WE WILL EXPLAIN NOW

First: prediction and imitation.
Then: preferences and rewards.

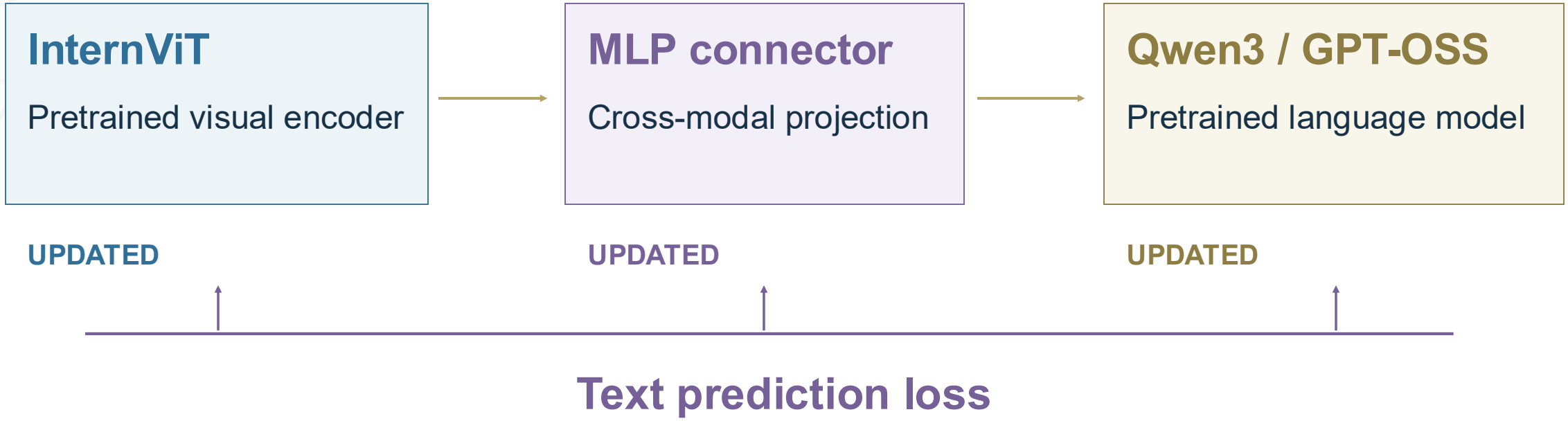
SFT belongs to post-training.
It supplies the starting model for Cascade RL.

The same image and question can be reused with a different training target at each phase.

InternVL3.5, Figure 3, §2.2 and §2.3. SFT belongs to post-training in the paper.

Native pretraining updates the complete VLM

During pretraining, update all three modules together using combination of large-scale text and multimodal corpora.



"Jointly trained" does not mean "initialized from scratch."

InternVL3.5, §2.2; native multimodal pretraining inherited from InternVL3.

The objective is still next-token prediction

$$\mathcal{L}_i = -\log p_\theta(x_i \mid x_{<i})$$

THIS REPRESENTS THE NEXT TOKEN PREDICTION (NTP) LOSS.
FOR CONVERSATIONAL SAMPLES: LOSS ONLY ON THE ASSISTANT RESPONSE TOKENS.

User + visual context

"Read the total on this receipt."

→

Assistant target

"The total is $7.50."

Context: no direct token loss

Response: contributes prediction loss

Visual features condition the text prediction; there is no pixel reconstruction target here.

What the model is asked to predict determines which parts of an example directly supervise learning.

InternVL3.5, §2.2, Equation 1. Training sample below is illustrative.

Square averaging: keep long answers from dominating

Long answers contain more target tokens. Should they dominate? During training, we need to alleviate bias toward either shorter or longer answers. This can be achieved via square averaging to reweight the NTP loss.

$$w_i = \frac{1}{\sqrt{N}}$$

Per-token weight
N = response length

$$N \times \frac{1}{\sqrt{N}} = \sqrt{N}$$

Total unnormalized response weight
Grows with √N, not N

100-token vs. 4-token response	Weight ratio	Interpretation
Equal token weighting	25 : 1	Length dominates
Square averaging	**5 : 1**	**Intermediate weighting**

Long answers still count more, but their contribution grows sublinearly with length.

InternVL3.5, §2.2, Equation 2. The 100-token / 4-token comparison is illustrative.

Pretraining data: broad sources, broad capabilities

116M samples

250B tokens

32K context limit

Multimodal sources

Mainly inherited InternVL3 corpora:
Image captioning, Q/A, math, science, charts,
OCR, knowledge grounding, documents understanding, dialogue, and medical.

Text-only sources

Primarily InternLM-series corpora, and further augmented with open-source datasets.

Reported text : multimodal mixture ≈ 1 : 2.5.

These are sample and token counts—not a count of unique images.

InternVL3.5, §2.2: pretraining data. Reported corpus sizes are approximate.

What does a multimodal training example look like?

Caption / scene description

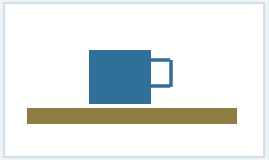


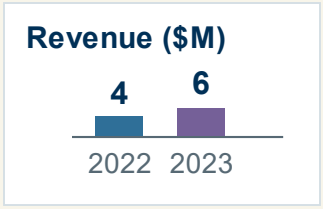
Image → description
“A blue cup on a table.”

OCR / document reading

RECEIPT
Tea $3.00
Cake $4.50

Question → answer
“Price of tea?”
“$3.00.”

Chart question answering



Question → answer
“Which year is higher?”
“2023.”

Visual mathematics

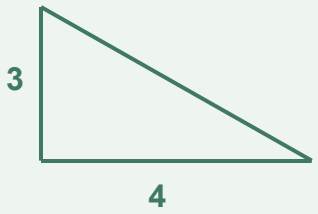


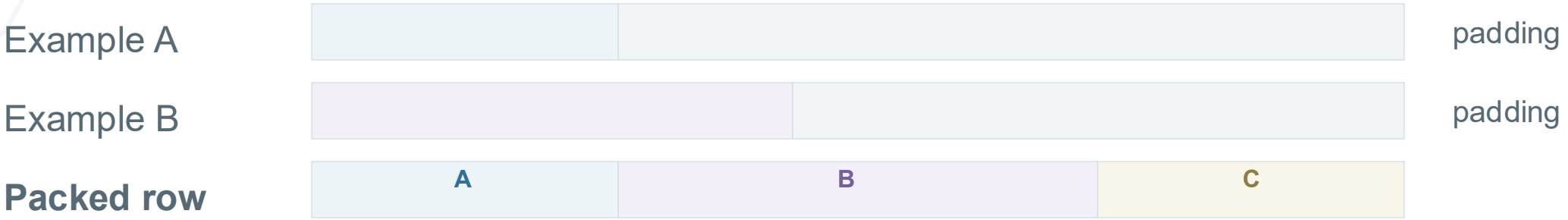
Diagram → solution
“Find the area.”
“½ × 3 × 4 = 6.”

The model sees different formats, but the supervision often arrives as text to predict.

Task families: InternVL3.5, §2.2. All four examples are illustrative—not released training items.

Supporting techniques keep training practical

PACK VARIABLE-LENGTH EXAMPLES MORE EFFICIENTLY



Packing reduces empty space; it does not create a new supervision signal.

Random JPEG compression

Expose training inputs to compression artifacts, improving model's real-world performance.

Systems support

32K context; distributed training and optimized kernels.

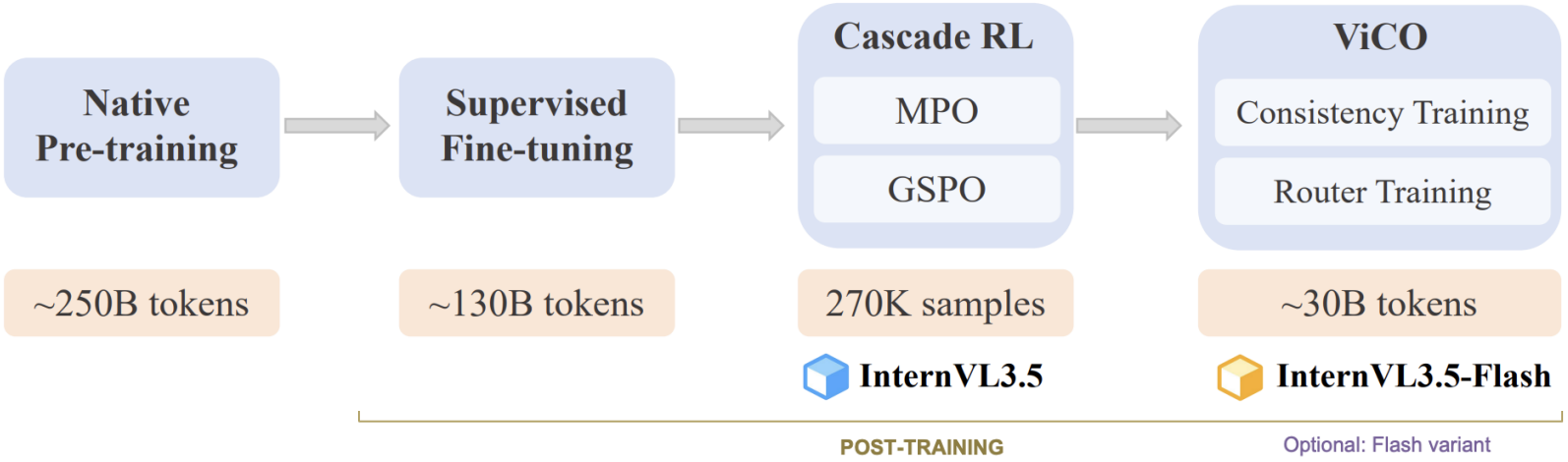
The data and objective teach the behavior; these techniques make the recipe more robust and tractable.

InternVL3.5, §2.2 and §2.5. Packing bars are schematic, not corpus measurements.

Post-training: after the architecture works

Architecture connects vision and language.
Post-training refines how the model uses that connection.

Overview | Cascade | MPO | GSPO



Three stages of post-training:

1. SFT — learn from better examples
Keep pretraining objective, but use higher-quality conversation data.

2. Cascade RL — improve reasoning
Start with offline preference learning, then refine with online RL.

3. ViCO — preserve performance with fewer visual tokens
Train for consistent outputs across visual compression rates, then learn when to use each rate through the visual resolution router (ViR).

LLM → TEXT-GENERATING VLM

$$\pi_\theta(y \mid q) \rightarrow \pi_\theta(y \mid I, q)$$

Same optimization; now grounded in an image.

Ouyang et al. (2022), InstructGPT; Wang et al. (2025), InternVL3.5, §2.3.

SFT: same loss, selected multimodal responses

TRAINING › SUPERVISED FINE-TUNING

56M	130B	1 : 3.5
samples	tokens	text-only : multimodal

HOW SFT WORKS

Imitate selected target responses.

Same next-token objective and
square averaging as in pretraining

WHAT THE TRAINING MIX ADDS

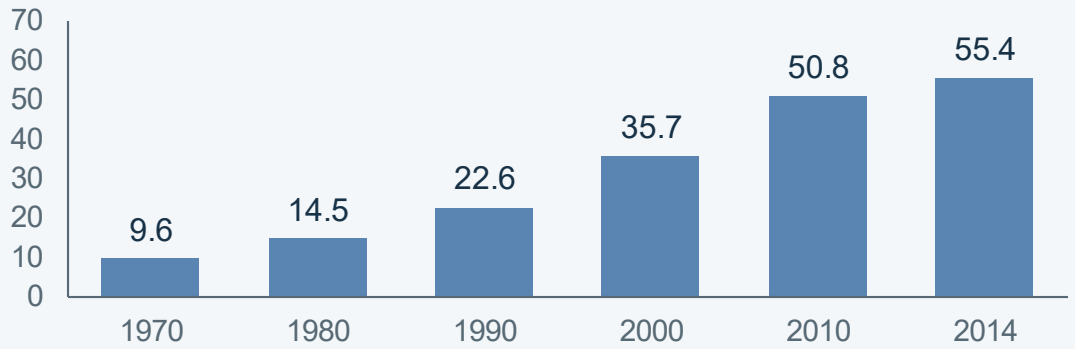
Inherited instructions
Broad task-following examples from InternVL3.

Reasoning traces
Detailed solutions from a larger reasoning model.

New capabilities
GUI, embodied-style, and SVG data.

Task-category weights are not reported.

EXAMPLE · CHARTQA QUESTION → TARGET



Q: What's the U.S. Hispanic population in 2000 (in millions)?

Target: **35.7**

ChartQA #10438

InternVL3.5, §2.3 (SFT + Data); public ChartQA training record 10438. Dataset sizes are approximate.

Overall RL recipe: Cascade RL

SFT
Starting policy

→

Offline: MPO
Reuse fixed comparisons: cheap, but limited.

→

Online: GSPO
Fresh attempts: adaptive, but expensive.

INTERNVL3.5-8B ABLATION

After SFT	Reasoning ↑	GPU hours
None	53.6	—
MPO	56.3	≈300
GSPO: 1 episode	57.3	≈5,500
GSPO: 2 episodes	58.2	≈11,000
MPO → GSPO	**60.3**	≈5,800

60.3 best reasoning score

≈½ the reported cost

versus the longer GSPO-only run

Online is not universally better: data coverage and reward quality still matter.

A cheap offline warm-up gives the online stage a better starting point.

Wang et al. (2025), InternVL3.5, Table 16; Song et al. (2024), The Importance of Online Data.

Offline RL: MPO(*Mixed Preference Optimization, 2024*)

Fixed examples: one prompt, one chosen response, one rejected response.

$$(x, y_c, y_r) = (\text{prompt}, \text{chosen}, \text{rejected})$$

x includes the image and question; c = chosen, r = rejected.

$$\mathcal{L}_{\text{MPO}} = w_p \mathcal{L}_{\text{DPO}} + w_q \mathcal{L}_{\text{BCO}} + w_g \mathcal{L}_{\text{gen}}$$

DPO · preference

Which answer is better?

Rank the pair.

BCO · quality

Positive or negative?

Classify each answer.

LM · generation

How do we write it?

Imitate the chosen answer.

A modular framework; InternVL3.5 inherits the original DPO + BCO + LM recipe.

One model receives the weighted sum of all three losses—not three separate networks.

Wang et al. (2024), Mixed Preference Optimization, §4.1; InternVL3.5, Eq. 3.

MPO continued: what each loss teaches the VLM

Implicit reward $u_j = \beta \log \frac{\pi_\theta(y_j \mid x)}{\pi_{\text{ref}}(y_j \mid x)}, \quad j \in \{c, r\}$

Intuitively: How much an answer is favored by the current model compared to the frozen reference.

LOSS	OBJECTIVE	WHAT IT TEACHES
DPO Preference loss	$\mathcal{L}_{\text{DPO}} = -\log \sigma(u_c - u_r)$	**Rank the pair** Max score diff between the chosen and rejected
BCO Quality loss	$\mathcal{L}_{\text{BCO}} = -\log \sigma(u_c - \delta) - \log \sigma(\delta - u_r)$	**Classify each answer** Put Chosen above δ; rejected below δ. δ computed from model implicit reward
LM Generation loss	$\mathcal{L}_{\text{gen}} = -\frac{1}{N} \log \pi_\theta(y_c \mid x)$	**Imitate the chosen answer** max likelihood of chosen

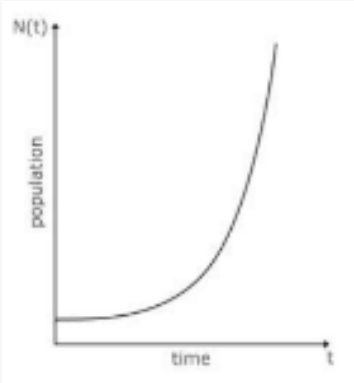
c / r: chosen / rejected • δ: moving baseline • N: chosen-response tokens • σ: sigmoid; β: score scale

VLM takeaway: an efficient offline warm-start before online GSPO.

Wang et al. (2024), MPO, §4.1; InternVL3.5 (2025), §2.3, Eq. 3. Per-pair losses; expectations omitted.

MPO data example: preference pairs across visual tasks

Published MMPR example



Which term matches the picture?

A. logistic growth

B. exponential growth

Original graph crop; prompt excerpt from Fig. 7(g).

✓ Chosen response · summary

Reads a J-shaped curve that keeps rising without leveling off.
Final answer: B — exponential growth.

✗ Rejected response · summary

Mistakenly describes an S-shaped curve that levels off.
Final answer: A — logistic growth.

MMPR-v1.2 mix: VQA + OCR/documents + charts + science/geometry + captioning

Release metadata uses source-specific repeat_time values; an overall text-only : multimodal ratio is not given.

Original MMPR example; graph cropped and responses summarized. Membership in the v1.2 training subset is unverified.

VLM interpretation: preference training must reward visual grounding, not only fluent reasoning.

InternVL3.5, §2.3; OpenGVLab/MMPR-v1.2, meta.json (source mix); MPO paper, Fig. 7(g) (example).

Online RL: GSPO (Group Sequence Policy Optimization, 2025)

For one image + question x, compare a group of G sampled responses y_i.

FULL GSPO OBJECTIVE · MAXIMIZE

InternVL3.5 · Eq. (5)

$$\mathcal{L}_{\text{GSPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\min(s_i(\theta)\,\hat{A}_i,\ \text{clip}(s_i(\theta),\,1-\varepsilon,\,1+\varepsilon)\,\hat{A}_i)\right]$$

$x \sim D;\ \{y_1, \ldots, y_G\} \sim \pi_{\text{old}}(\cdot \mid x).$ π_{old} = rollout VLM; π_θ = current VLM.

Minimized loss: $-\mathcal{L}_{\text{GSPO}}$.

1 GROUP ADVANTAGE

$$\hat{A}_i = \frac{r_i - \text{mean}_j\, r_j}{\text{std}_j\, r_j}$$

$r_i = r(x, y_i)$; j runs over the same group.

Above mean → reinforce.
Below mean → discourage.

2 SEQUENCE RATIO

$$s_i = \left(\frac{\pi_\theta(y_i \mid x)}{\pi_{\text{old}}(y_i \mid x)}\right)^{1/|y_i|}$$

$|y_i|$ = number of generated tokens.

Geometric mean of token ratios; one weight for the whole response.

3 CLIPPED INCENTIVE

$\hat{A}_i > 0$: $s_i > 1 + \varepsilon$

No extra incentive to increase it.

$\hat{A}_i < 0$: $s_i < 1 - \varepsilon$

No extra incentive to decrease it.

ε sets the band; not a hard update bound.

Compare answers within a query; update and clip at the response level.

Wang et al. (2025), InternVL3.5, §2.3, Eqs. 4–6; GSPO: Zheng et al. (2025), §4.1.

GSPO data: difficulty-filtered, mixed-source prompts

DATA › CASCADE RL › GSPO — CITED DEEPSCALER SOURCE

DeepScaleR source example

PROBLEM · PUBLIC DATASET RECORD

$$991 + 993 + 995 + 997 + 999 = 5000 - N$$

Find N.

Recorded answer: 25

Sum: $5 \times 995 = 4975$

$N = 5000 - 4975 = 25$

Arithmetic check shown here—not a logged model rollout.

How the prompt pool is mixed

Retain queries with 0.2–0.8 accuracy

Measured from existing MMPR-v1.2 rollouts

↓

Add diverse reasoning sources

Visual: MM-Eureka / R1-Onevision

Text-only: DeepScaleR (example at left)

↓

MMPR-Tiny · about 70K queries

Mix visual grounding with text reasoning; then generate new responses online.

Exact source weights and modality proportions are not reported.

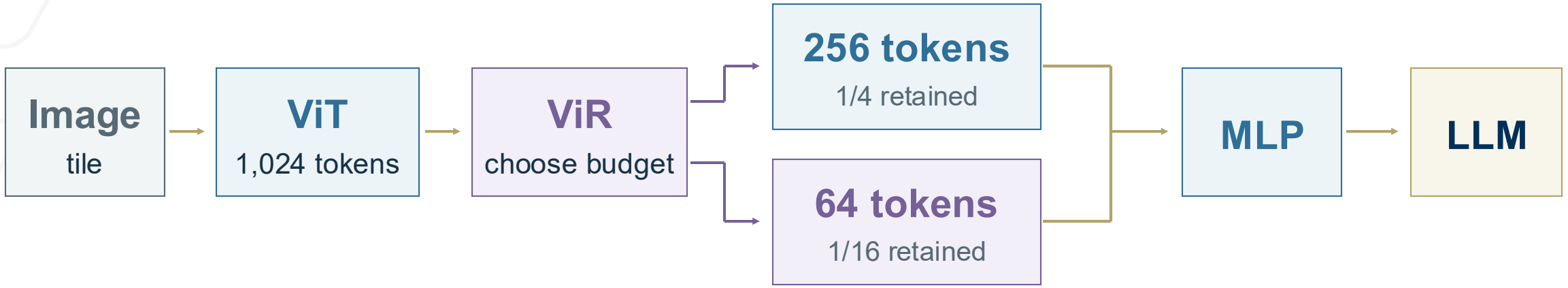
DeepScaleR is a cited source; inclusion of this text-only example in MMPR-Tiny is unverified. No logged rollouts shown.

The prompt mixture is curated; the responses and their rewards are produced during online training.

InternVL3.5, §2.3 Data, refs. 90/163/83; MMPR-Tiny card; public DeepScaleR source example.

Visual Consisency learning (ViCO): compressing token

Flamingo: fixed 64 tokens/image or video. Flash: 64 or 256 tokens/tile.



Tile = one 448 × 448 crop. Each route uses pixel shuffle before the MLP.

STAGE 1 · CONSISTENCY TRAINING

Frozen reference: 256 tokens → p

Student: sample 64 / 256 tokens → q

Same image, question and answer prefix.

MATCH NEXT-TOKEN DISTRIBUTIONS

$$\mathcal{L}_{\text{ViCO}} = \mathbb{E}_{k\in\{64,256\}}\left[\frac{1}{T}\sum_{t} D_{\text{KL}}\left(p_t^{256} \,\|\, q_t^{k}\right)\right]$$

Train ViT, MLP and LLM; keep the reference frozen.

Preserve the answer distribution—not the pixels or visual features.

Wang et al. (2025), InternVL3.5, §2.1 / §2.3, Eq. 7; Alayrac et al. (2022), Flamingo, §2.1.

Visual Resolution Router (ViR): learn the routing decision

How do we decide which tiles need the larger token budget?

CONSISTENCY-LOSS RATIO

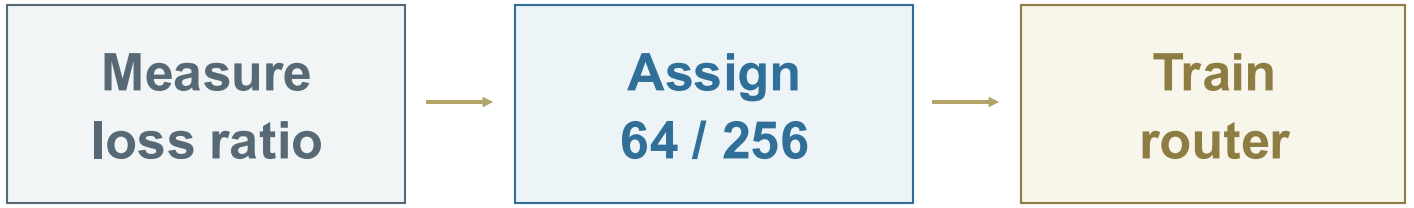
$$r = \frac{\mathcal{L}_{64}}{\mathcal{L}_{256}}$$

$$r < \tau \rightarrow 64 \text{ tokens}$$

$$r \geq \tau \rightarrow 256 \text{ tokens}$$

τ: dynamic threshold from recent loss ratios

BUILD LABELS, THEN TRAIN ONLY ViR



Freeze ViT, MLP, and LLM.
Train the router with binary cross-entropy.

Inference: visual features → router → selected token budget

Compare losses during training; at inference, the learned router chooses directly.

Wang et al. (2025), InternVL3.5, §2.3, Eqs. 8–9. Simplified notation; architecture schematic.

Result: fewer tokens, a small quality drop (InternVL3.5 Flash)

TABLE 17 · SELECTED OVERALL SCORES

Model	Original	Flash	Change
InternVL3.5-8B	80.2	79.8	−0.4
InternVL3.5-38B	83.9	83.4	−0.5

≈50%

fewer visual tokens

Token count—not end-to-end latency

DATA REUSE · DIFFERENT SUPERVISION

Consistency training

Primarily the same SFT datasets

Target: full-budget teacher distribution

Router training

Subset emphasizing OCR / VQA

Target: compression-sensitivity labels

Full original-vs-Flash comparison—not an isolated ViCO ablation.

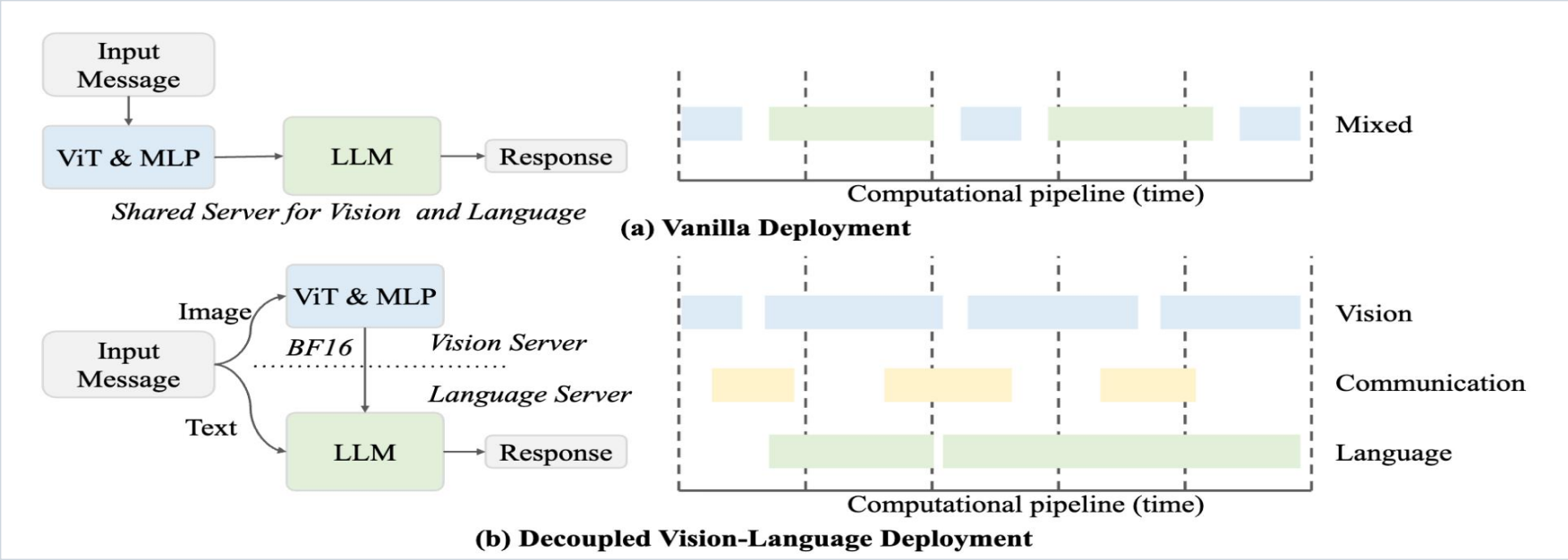
Roughly half the visual tokens, with a small drop on the selected aggregate.

Wang et al. (2025), InternVL3.5, Table 17 and §2.1 / §2.3. Scores copied from the v2 table.

DvD: decoupled vision–language deployment

INFERENCE EFFICIENCY › DVD

InternVL3.5, §2.5 · Fig. 4 + Table 18



Mechanism

1. Split serving into a vision server (ViT + MLP, optionally ViR) and a language server (LLM prefill + decoding).
2. The vision side processes image inputs and sends compact BF16 visual features to the language side.
3. Requests are pipelined: vision for request n+1 can overlap language generation for request n.

Result table · Table 18 (38B model, resolution 896)

Setting	Req/s	Speedup
Baseline	2.71	1.00×
+ DvD	5.06	1.87×
+ DvD + ViR	10.97	4.05×

DvD increases efficiency by overlapping vision and language across requests; with ViR, throughput rises from 2.71 to 10.97 requests/s (4.05×).

51

Wang et al. (2025), InternVL3.5, §2.5 and Fig. 4 / Table 18. Official illustration from the paper is used above.

Read the benchmarks as tasks, not a leaderboard

Read fine detail	Use visual evidence	Understand time	Act in a GUI

MMMU

College-level questions across disciplines.

Needs subject knowledge plus the ability to interpret diagrams, tables, or images.

MathVista

Mathematics grounded in visual inputs.

Needs correct visual extraction before computation and reasoning.

Selected results below are the report's August 2025 evaluations—not a current leaderboard.

Two benchmarks anchor the discussion; the task determines what a score means.

InternVL3.5, §3; Yue et al., MMMU (2023); Lu et al., MathVista (2023).

The complete recipe improves visual reasoning

Model	MMMU (val)	MathVista (mini)	7-test aggregate
InternVL3-8B	62.7	71.6	44.3
InternVL3.5-8B	**73.4**	**78.4**	**60.3**
InternVL3.5-241B-A28B	77.7	82.7	66.9

+16.0 points

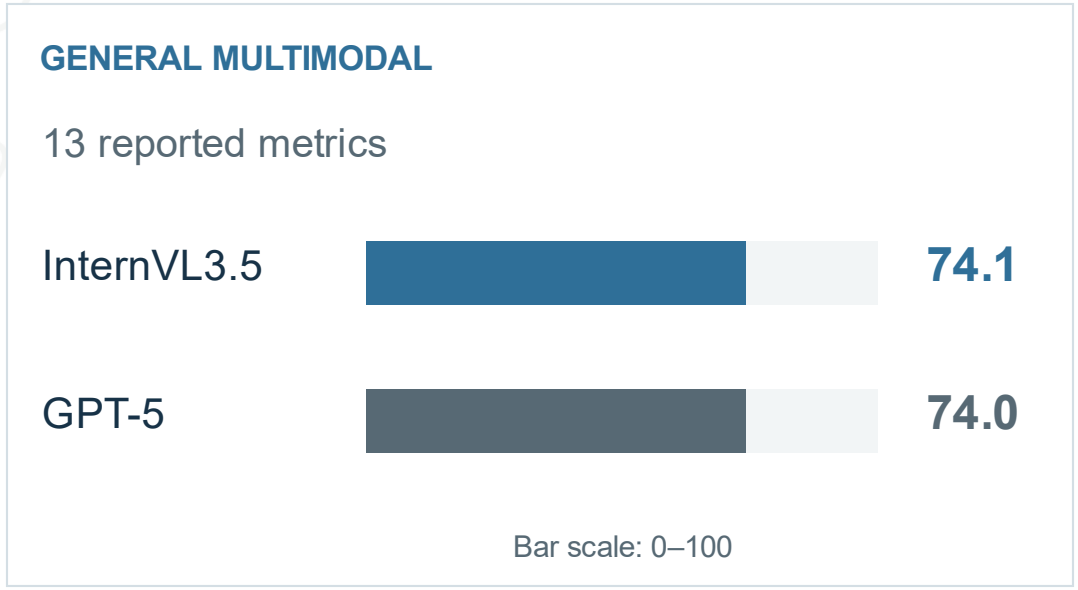
8B aggregate: 44.3 → 60.3.
A full-recipe gain—not an RL-only effect.

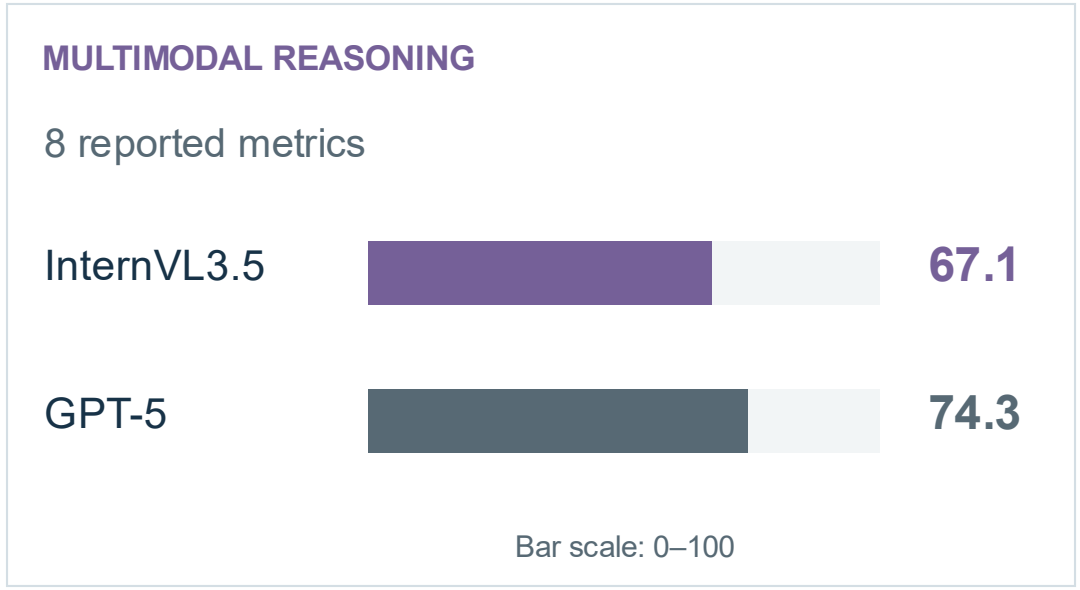
Same nominal size does not hold backbone, data, or training recipe constant.

InternVL3.5, Table 3. Default rows, without the separate Parallel Thinking setting.

Comparison with GPT-5: general vs specific score

InternVL3.5-241B-A28B versus the GPT-5 column in the paper.





A +0.1 general aggregate does not erase a −7.2 reasoning gap.

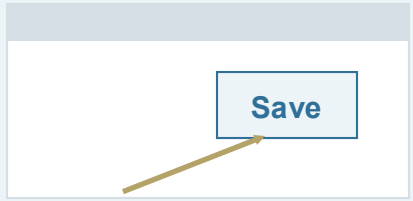
Read the task group and evaluation protocol before turning one aggregate into a headline.

InternVL3.5, Table 2. Historical 2025 report; the table combines several score-source protocols.

Still way to go

Locate the target

ScreenSpot-v2: static GUI grounding



92.9

Is the requested UI element correctly located?

Finish the task

WindowsAgentArena: online GUI interaction

18.0

Does the full action sequence succeed?

Largest model; Table 10 uses a 50-step budget for this online result. Different benchmarks, different metrics.

Good perception is necessary for agents, but sustained control and recovery remain additional challenges.

InternVL3.5, Table 10 and §3.11–§3.12. Values are from different tasks, not a matched gap experiment.

What is strongest about the paper?

A complete, reusable system	One visual-language interface across many task families.
Targeted evidence for key additions	Cascade RL and serving comparisons test concrete design choices.
Quality and deployment are considered together	Flash exposes an explicit token-cost / quality tradeoff.

The strongest contribution is the full recipe, supported by targeted ablations.

Synthesis based on InternVL3.5, §2–§3 and Tables 16–18; not a new experiment.

What the results still do not settle

Uneven gains

More reasoning training does not automatically improve every perception or hallucination test.

Causal attribution

Backbone, data, scale, and objectives change together. A large benchmark gain is not an explanation of its cause.

Reproducibility

Checkpoints are easier to reuse than the entire data, filtering, and reward pipeline is to reproduce.

Metric boundaries

Average scores can hide hard cases. Spatial QA is not physical control; throughput is not latency.

Broad coverage is valuable, but it is not proof of universal grounding or robust real-world behavior.

InternVL3.5, §3.6–§3.7 and §2–§3; attribution and reproducibility points are presenter critique.

The source of adaptation shifts across the papers

Learning source	Flamingo	InternVL3.5
Broad pretraining	Web documents + paired captions	Text + multimodal corpora
Task behavior	Examples in the prompt	SFT instructions + reasoning traces
Preference / reward	No such stage in the main recipe	MPO pairs → GSPO sampled responses
Visual compression	Resampler trained with language loss	ViCO reuses SFT; router uses OCR/VQA

Flamingo also tests supervised task fine-tuning; the contrast concerns the main training recipe.

The loss matters, but so does which examples and feedback reach the model.

Alayrac et al. (2022), Flamingo, §2.4–2.5; §3.2; Wang et al. (2025), InternVL3.5, §2.2–2.3, Data

Data mixture: Flamingo vs InternVL3.5

DATA › MIXTURE COMPARISON › SAMPLE FORMAT + TRAINING RECIPE

Flamingo is stream-based; InternVL3.5 is curriculum-based.

Flamingo (2022)

four explicit streams mixed by weighted gradients

TRAINING EXAMPLES

M3W: text ... <image> ... text ... <image> ... text
256-token webpage span; up to 5 images
ALIGN/LTIP: image + short/long text; VTP: video + text

MIXING OPERATION

separate dataset batches -> separate gradients
weighted gradient sum -> one update
ablation: better than round-robin

WHAT LEARNS?

Frozen Chinchilla LM.
Train visual bridge/resampler/cross-attn to inject visual context.

InternVL3.5 (2025)

text-only + multimodal examples under NTP

PRETRAINING MIXTURE

116M samples / 250B tokens; text-only : multimodal ~= 1 : 2.5
multimodal pool: captions, VQA, OCR, charts, science, documents, grounding, dialogue

SFT MIXTURE

56M samples / 130B tokens; text-only : multimodal ~= 1 : 3.5
instruction data + filtered thinking traces + capability-expansion data

WHAT LEARNS?

ViT + MLP + LLM are jointly updated before RL/ViCO.
Training builds a full VLM, not just an adapter.

Caveat: InternVL3.5 reports modality-level ratios and task families, but not exact per-dataset sampling weights.

Flamingo: stream mixture for a frozen LM. InternVL3.5: curriculum mixture for a fully trained VLM.

Progress is broader than one leaderboard number

Flamingo asks…

Can a few prompt examples adapt one model across image and video tasks?

Evidence: shot curves and held-out tasks.

InternVL3.5 asks…

Can a post-trained VLM improve reasoning and remain efficient to serve?

Evidence: RL, Flash, and deployment ablations.

A useful next experiment

Match data, backbone, and visual-token budget; then change the interface or freezing policy.

Compare design claims under their protocols—not incomparable aggregate scores.

Alayrac et al. (2022), Flamingo, §3; Tables 1–3; Wang et al. (2025), InternVL3.5, §3; Tables 16–18

Limitations become deployment questions

Issue	Why the stakes grow	Proposed safeguard
Grounding and hallucination	Fluent answers may miss visual evidence.	Verify against the image; allow abstention.
Web-data bias and privacy	Training inherits uneven coverage and sensitive content.	Audit subgroups and data provenance.
From answers to actions	GUI use can turn a wrong answer into a wrong action.	Constrain tools; confirm consequential steps.

Safeguards are proposed extensions, not validated features of either paper.

More capability raises the need for grounding checks, oversight, and clear scope.

Alayrac et al. (2022), Flamingo, App. D.1–D.2; Wang et al. (2025), InternVL3.5, §3.12–3.15

Discussion: what should a general VLM preserve?

01 **When should backbones stay frozen?**

What changes when more multimodal data and compute become available?

02 **What is safe to compress?**

Does preserving language answers also preserve geometry needed for action?

03 **Where should new task learning happen?**

In prompt examples, parameter updates, or both?

The common problem is preserving useful knowledge while adapting the system.

Alayrac et al. (2022), Flamingo, App. D.1; Wang et al. (2025), InternVL3.5, §2.1–2.3