LLaVA and Qwen3-VL

Visual Instruction Tuning (LLaVA)
Liu, Li, Wu & Lee (2023)

Qwen3-VL Technical Report
Qwen Team (2025)

Yuzhou Wang, Emile Anand, Ijay Narang

Georgia Tech

Outline

01

Problem & foundations

Why visual instruction following?

02

LLaVA

Data generation, model, training and evidence

03

Qwen3-VL

Architecture, training and broader capabilities

04

Comparison & discussion

Evaluation, performance analysis, and key takeaways

Georgia Tech

Recognizing objects is not yet following an instruction

The image supplies evidence; the user's question determines which evidence matters and what kind of answer to produce.

THE SAME IMAGE, A DIFFERENT TARGET

"Describe the scene" and "What is unusual?" require different answers from the same pixels.

Question: "What is unusual about this image?"

1. **Recognize**

   The scene contains a person, clothing, an ironing board, and a vehicle.

2. **Relate**

   The activity is ironing; the setup is on a vehicle rather than in an ordinary indoor workspace.

3. **Answer the instruction**

   Explain that unexpected activity–setting combination. Merely naming the objects misses the request.

LLaVA: teach this behavior with generated instructions. **Qwen3-VL:** extend the scope to detail, long context, and actions.

Visual instruction following = relevant visual evidence + the requested response behavior—not an object list.

Liu et al., §1 and Table 3, pp. 1–2, 6

Georgia Tech

LLaVA: a simple model—and supervision that teaches behavior

Large Language and Vision Assistant: reuse pretrained capabilities, then teach the visual interface and the interaction style.

1. A deliberately lightweight visual interface

CLIP ViT-L/14 → linear W → Vicuna

Representation. Project a sequence of visual features into the language model's embedding width; do not first convert the image into a caption.

2. A data-generation solution

Captions + boxes → GPT-4 → instructions

Supervision. A text-only teacher turns existing annotations into questions and answers. The student learns from the original image and these generated targets.

158K INSTRUCTION SAMPLES • APPROXIMATELY 80K UNIQUE IMAGES

58K — **Conversation**

Focused answers and multi-turn follow-up.

23K — **Detailed description**

Broad coverage of entities and relationships.

77K — **Complex reasoning**

Explanations that connect evidence to implications.

The connector makes visual evidence accessible; the instruction dataset teaches what a useful response should do.

Liu et al., §§3–4, pp. 3–5; §5.1, p. 6

Georgia Tech

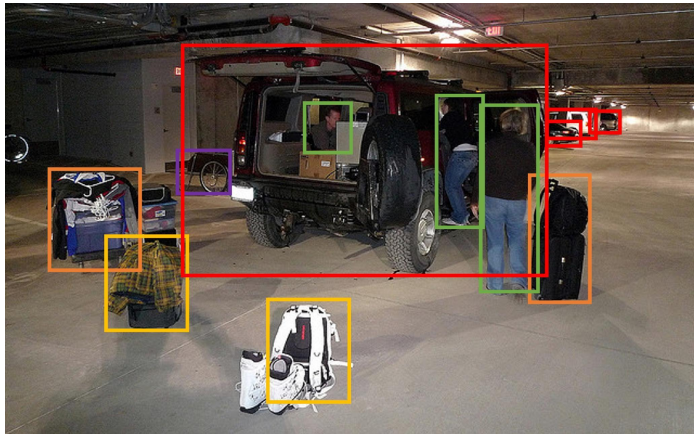
A caption can name the scene without answering the question

Rephrasing “describe this image” does not automatically create supervision for localization, comprehensive description, or reasoning.

Cheap caption expansion

Human: Describe the image.

Assistant: People try to fit their luggage in an SUV.



Three instructions, three different targets

“Where is the vehicle parked?”

Select a specific fact: the underground parking area.

“Describe the objects and their positions.”

Cover people, luggage, the SUV, and their spatial relations.

“What challenges do the people face?”

Explain packing constraints using the scene and practical knowledge.

Captions are useful for initial alignment; richer instruction–answer pairs teach the assistant to change its response with the task.

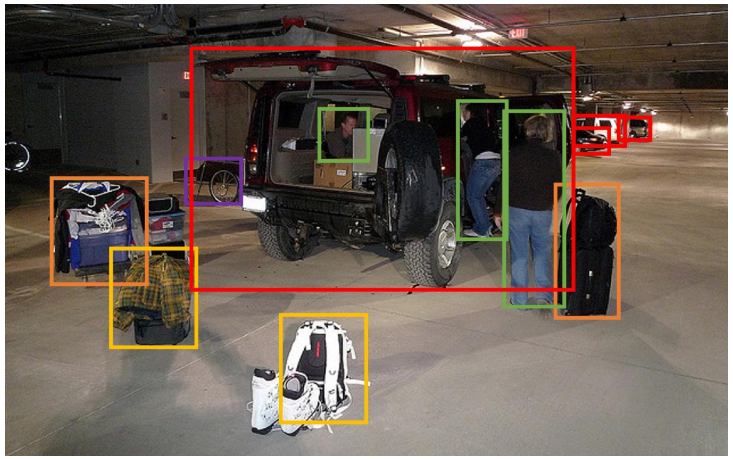
Liu et al., §3, pp. 2–4; Table 14, p. 23

Georgia Tech

The teacher receives annotations; the student receives pixels

LLaVA generates visual instruction data without giving the GPT-4 teacher an image input.

The image being described



Boxes name objects and locate them; captions describe the scene and activity.

An offline supervision pipeline

Caption context. People are loading luggage into an SUV in an underground parking area.

```
person: [0.681, 0.242, 0.774, 0.694]
```

Normalized box coordinates encode spatial location, not the appearance of every pixel.

GPT-4 teacher

Annotations + a few seed examples

→

Generated supervision

Question + target response

LLaVA training: original image + generated question → predict the generated answer.

Captions and boxes are a generation-time scaffold—not extra annotation inputs required by the trained LLaVA assistant.

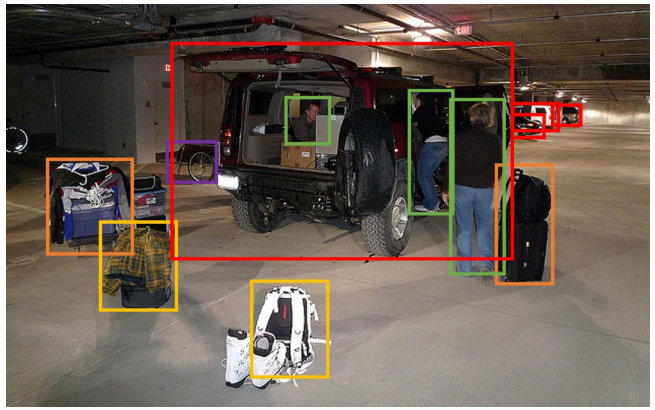
Liu et al., §3; Tables 13–14, pp. 22–23

Georgia Tech

Conversation teaches focused answers across turns

One image supports several questions: the answer must change with the request while remaining consistent with the shared scene.

Shared visual context



Prompt coverage. Object types, counts, actions, locations, and relative positions.

Generate answerable, grounded questions.

A multi-turn training example

1 **What type of vehicle is in the image?**

A black sport utility vehicle. — **Identify**

2 **Where is the vehicle parked?**

In an underground parking area. — **Localize**

3 **What are the people doing?**

They are trying to fit their luggage into the SUV. — **Describe action**

The image remains in context; each assistant response supplies a prediction target.

A conversation is not a set of unrelated captions: each answer conditions on the image, the new question, and the preceding dialogue.

Liu et al., §3 and §4.2; Tables 2, 13–14

Georgia Tech

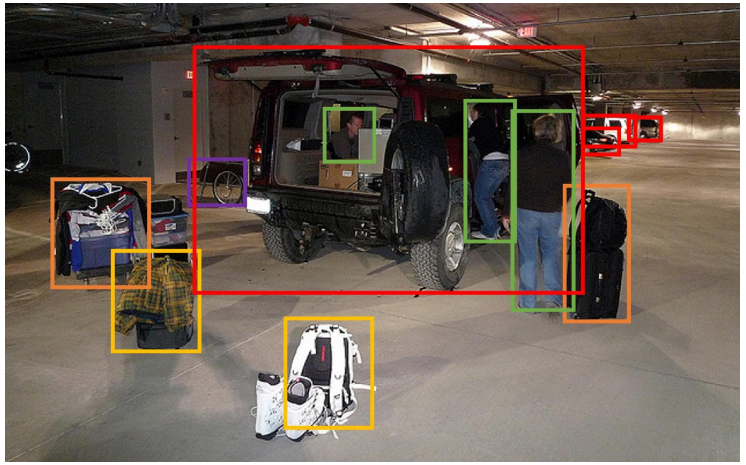
Detailed descriptions teach coverage and organization

Instead of answering one narrow question, the assistant must build a coherent account of the scene and its relationships.

One comprehensive response

SINGLE-TURN

“Describe the image in detail.”



Organize the answer from scene to relations

Scene and main entity

Start with the underground parking area and the black SUV.

People and activity

Describe the people around the vehicle and their luggage-loading activity.

Objects and spatial relations

Include surrounding luggage and nearby vehicles, locating objects relative to the SUV.

Interpretation: coverage, ordering, and relation words are the added supervision—not a new visual encoder.

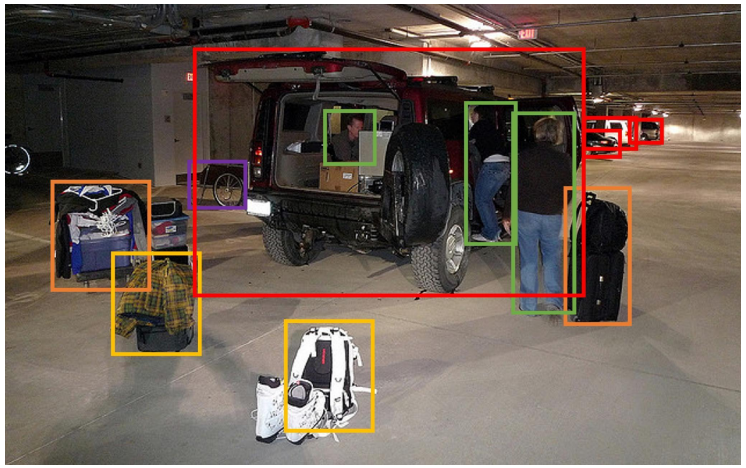
Detail is useful when it adds grounded coverage and relationships—not when it merely makes the response longer.

Liu et al., §3; Table 14, p. 23

Complex reasoning connects observations to implications

The response explains what follows from a scene, while separating visible evidence from assumptions and practical knowledge.

What challenges do they face?



The question asks for an explanation—not simply "people, luggage, SUV."

A grounded explanation has distinct parts

1 **Observation**

People and several pieces of luggage are gathered around one SUV.

2 **Relevant constraints**

A vehicle has limited space; passengers need room, and the driver needs visibility.

3 **Reasoned implication**

They may need to arrange the luggage carefully to fit it without crowding passengers or blocking the view.

"May need to" marks an inference. Exact luggage volume, capacity, and plans are not established by this example.

Grounded reasoning uses the image as a premise; it should not present every plausible implication as an observed fact.

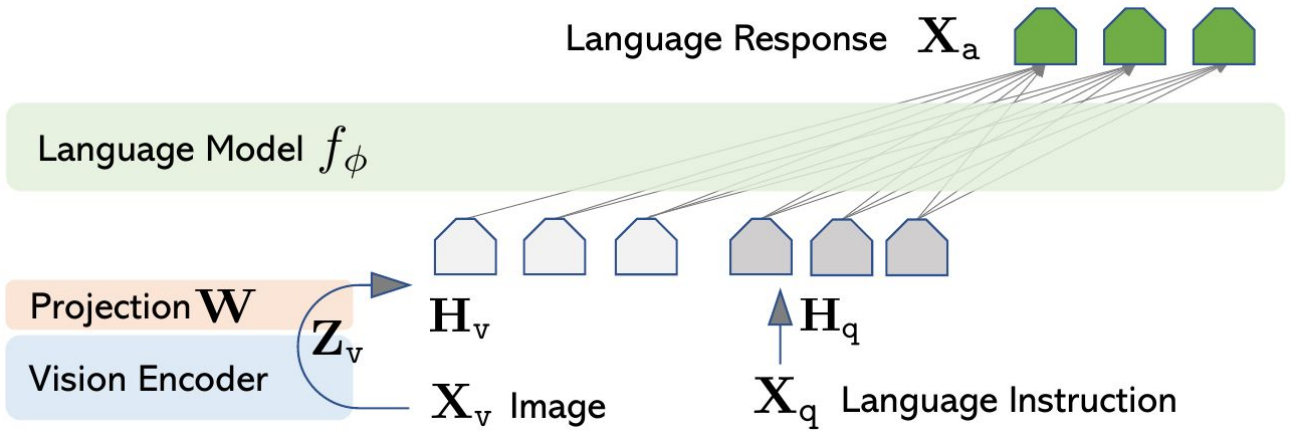
Liu et al., §3; Table 14, p. 23 • decomposition is explanatory

Georgia Tech

LLaVA architecture: visual vectors become decoder inputs

CLIP supplies a grid of features; a linear map changes their width; Vicuna generates a response conditioned on image and instruction.

Trace the two input paths



$$Z_v = g(X_v), \qquad H_v = WZ_v$$

$$Z_v \in \mathbb{R}^{d_v \times N_v}, \quad W \in \mathbb{R}^{d_L \times d_v}, \quad H_v \in \mathbb{R}^{d_L \times N_v}$$

What each component does

CLIP ViT-L/14. Encode the image into N_v visual vectors. These are learned features, not necessarily one object per vector.

Projection W. Apply the same learned map to every visual vector, changing the feature width to Vicuna's embedding dimension.

Vicuna decoder. Attend to visual and text embeddings together, then predict the answer one token at a time.

The linear projection aligns feature dimensions. It does not produce a caption, add a Q-Former, or compress the number of visual tokens.

Liu et al., §4.1, Figure 1, Equation (1), p. 4

Georgia Tech

Align the visual interface while both backbones stay frozen

Learn W using image–caption pairs before asking the language model to adapt to complex multimodal interactions.

Only the projection is trainable

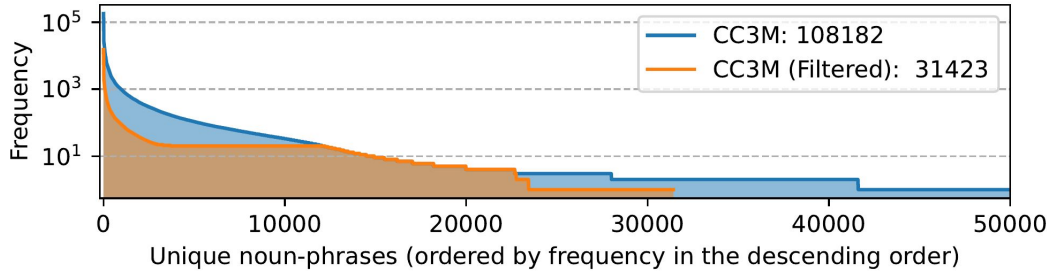
CLIP — Frozen → **W** — Train → **Vicuna** — Frozen

$$\min_{W} -\sum_{t} \log p_{W,\phi_0}(c_t \mid X_v, q, c_{<t})$$

c = original caption; q = a brief-description request.
φ_0 stays fixed; gradients still pass through the decoder to W.

PAIRS	EPOCHS	BATCH	LEARNING RATE
595K	1	128	2×10^{-3}

Balance concept coverage and training cost



Filtered CC3M. Reduce the dominance of frequent concepts while retaining broad coverage in a smaller set of image–text pairs.

Figure 7 counts noun-phrase frequencies. Its legend is not a count of training images.

Alignment is learned through caption prediction—not direct regression onto a precomputed "correct" word embedding.

Liu et al., §4.2; §5; Appendices C and E, Figure 7

Georgia Tech

Adapt W and Vicuna—and supervise the assistant's answers

"End-to-end" in this paper does not mean every component updates: the CLIP vision encoder remains frozen.

Two task-specific fine-tuning branches

Trainable parameters. Update θ = {W, φ}; retain the frozen CLIP encoder.

Multimodal chat

158K generated examples; 3 epochs.
Learn conversation, detailed description, and reasoning.

ScienceQA

Separate task-specific training.
Generate reasoning, then a multiple-choice answer.

One sequence, selective prediction targets

SCHEMATIC MULTI-TURN SERIALIZATION

Image + user Q_1	**Assistant A_1**
User Q_2	**Assistant A_2**

$$\mathcal{L}(\theta) = -\sum_{i \in \mathcal{A}} \log p_\theta(x_i \mid X_v, x_{<i})$$

A marks assistant-target positions (and stop behavior).
Past questions and answers remain conditioning context.

Stage 1 teaches the interface; Stage 2 changes the response policy. Questions provide context, while assistant answers supply the loss.

Liu et al., §4.2, Table 2; §5; §5.2

Georgia Tech

GPT-4 supplies a reference—and a quality judgment

The candidate sees pixels; the reference and evaluator use textual image annotations. These are different information channels.

01 Candidate answer

LLaVA receives the image and question, then generates an answer.

02 Reference answer

Text-only GPT-4 receives annotations and the question, then produces a reference.

03 Quality judgment

GPT-4 rates both answers from 1–10 for helpfulness, relevance, accuracy, and detail, using the annotations as context.

Interpret the relative score

$$100 \times \frac{\text{candidate quality score}}{\text{reference quality score}}$$

Illustration: a score of 8 versus 9 gives 88.9—not "88.9% correct."

What this evaluation can and cannot establish

Useful for broad, open-ended answers.
Dependent on annotation coverage and model judgments.
Not an equal-information comparison with a pixel-based GPT-4.

A relative judge score measures rated response quality against a reference. It is not a percentage of correctly answered questions.

Liu et al., §5.1, pp. 6–7 • metric example is illustrative

Georgia Tech

The full instruction mixture improves all three response types

COCO relative judge scores: the supervision mixture and its amount change—not the underlying architecture.

LLaVA-Bench (COCO): 30 images, 90 questions

Training data	Conv.	Detail	Reason	All
No instruction tuning	22.0	24.0	18.5	21.5
Conversation only	76.5	59.8	84.9	73.8
Conv. + small additions	81.0	68.4	91.5	80.5
Detail + reasoning	81.5	73.3	90.8	81.9
Full three-style mixture	**83.1**	**75.3**	**96.5**	**85.1**

Small additions = 5% of detail + 10% of reasoning examples, on top of conversation data.

How to read the gain

+11.3

Overall: 73.8 → 85.1

Detail gains +15.5. The full mixture improves 59.8 to 75.3.

Small additions help. Overall rises to 80.5 before using the entire mixture.

Richer supervision helps in this setup, but mixture diversity and training-data volume are not independently controlled.

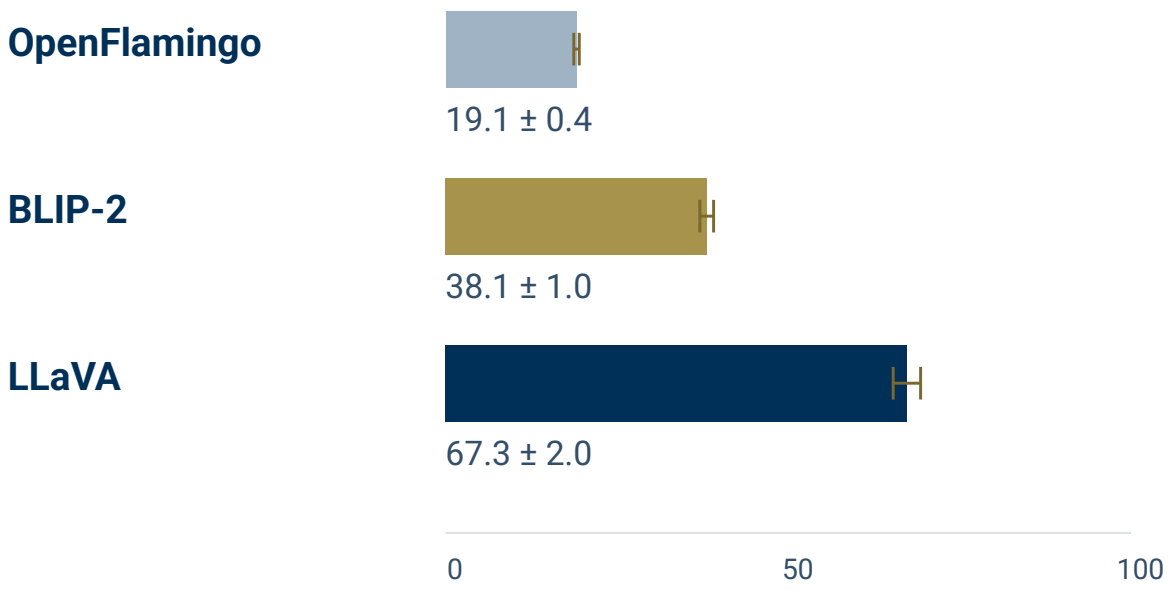
Liu et al., Table 4, p. 7 • deltas computed from displayed scores

Georgia Tech

In-the-Wild: useful transfer, measured on a small benchmark

Twenty-four images and sixty questions include varied scenes, memes, paintings, and sketches rather than COCO photos alone.

Overall relative score (mean ± SD)



Three inference runs; error bars show the reported SD.

Where the gains appear—and the caveats

57.3	52.5	81.7
Conversation	**Description**	**Reasoning**

Interpretation. The strongest reported category is complex reasoning, consistent with richer explanatory supervision.

Boundary. Only 24 images; a model judge; annotation-informed references; specific tested checkpoints and prompts.

These results support broader instruction-following behavior—not a universal ranking or a guarantee of robust visual grounding.

Liu et al., Table 5 and §5.1, p. 7

Georgia Tech

92.53% is the ensemble result—not standalone LLaVA

Unlike the open-ended judge score, ScienceQA reports multiple-choice answer accuracy on 4,241 test questions.

Separate fine-tuning and a different metric

Model / combination	Accuracy
GPT-4 (text only)	82.69%
LLaVA alone	90.92%
LLaVA + GPT-4 complement	90.97%
LLaVA + GPT-4 reconsideration	**92.53%**

+1.61 percentage points over standalone LLaVA

Task-specific ScienceQA training; some questions have no image context.

Ensemble logic: reconsider disagreement

Two initial answers

LLaVA prediction + text-only GPT-4 prediction

↓

When they disagree, GPT-4 receives the question and both outputs, then selects a final answer.

Why it can help. Language knowledge can correct some mistaken inferences, even though GPT-4 cannot independently inspect the image.

GPT-4 helps choose the final answer in the ensemble; ground-truth labels—not GPT-4's quality rating—determine accuracy.

Liu et al., §5.2, pp. 8–9; Table 7

Georgia Tech

Object co-occurrence is not attribute grounding

LLaVA can follow the requested answer format and still bind the wrong attribute to the wrong object.

“Is there strawberry-flavored yogurt?”

Reported failure: LLaVA answers yes.

Recognized objects do not establish the relation

Supported premise. Strawberries and yogurt are both present.

strawberries + yogurt ≠ strawberry-flavored yogurt

Missing evidence. The flavor must be linked to a particular yogurt container—for example by correctly reading its label.

Bridge to Qwen3-VL. Richer visual features and finer-grained supervision target such evidence bottlenecks. They are not proof that this exact failure is solved.

Instruction tuning teaches useful interaction; reliable answers still require the right evidence, relations, and uncertainty.

Liu et al., §5.1 / Table 6; Qwen3-VL §§1–2 • interpretation labeled

Georgia Tech

Qwen3-VL: from pixels to reasoning and action

A strong language model is the starting point—not a capability to sacrifice when adding vision.

How do we add rich visual understanding, long context, and useful actions—while preserving language ability?

MODEL FAMILY

Dense: 2B / 4B / 8B / 32B
MoE: 30B-A3B / 235B-A22B

Visual inputs
Images, video, documents

→

Vision encoder
Dynamic-resolution features

→

Merger + DeepStack
Compress and transfer

→

Qwen3 decoder
Reason, ground, use tools

THREE ARCHITECTURAL UPGRADES

DeepStack
Inject multiple ViT depths.

Interleaved MRoPE
Balance position frequencies.

Text timestamps
Make elapsed time explicit.

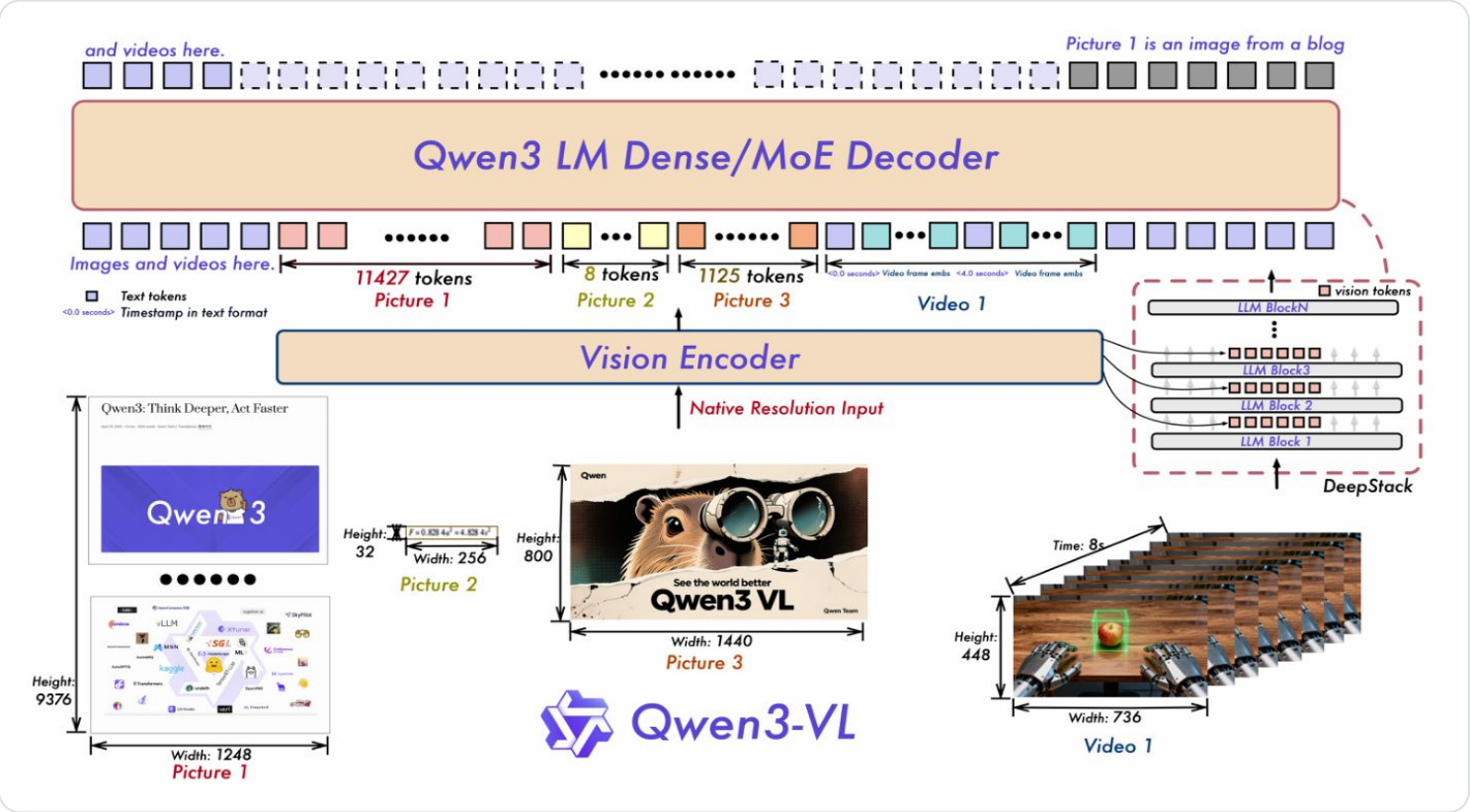
The system goal: strong perception + reliable information transfer + 256K multimodal context + effective training.

Report §§1–2, pp. 2–4

Georgia Tech

The familiar skeleton, upgraded at the interfaces

Figure 1 is best read as two visual routes: the main token stream and the additional cross-layer injections.



1 Encode the visual input

A SigLIP-2-derived ViT produces a variable-length spatial feature sequence.

2 Compress and align

A two-layer MLP maps each 2 × 2 feature group to one vector of LLM hidden width.

3 Fuse inside the decoder

Main visual vectors join the text sequence. Three intermediate ViT feature sets also enter early LLM layers.

A visual token is a continuous feature vector, not necessarily a word, an object, or a caption.

Report Fig. 1 and §2, pp. 2–4

Georgia Tech

Preserve image detail, then compress before the LLM

The ViT and the language model need not operate on the same number of tokens.

Dynamic-resolution visual encoder

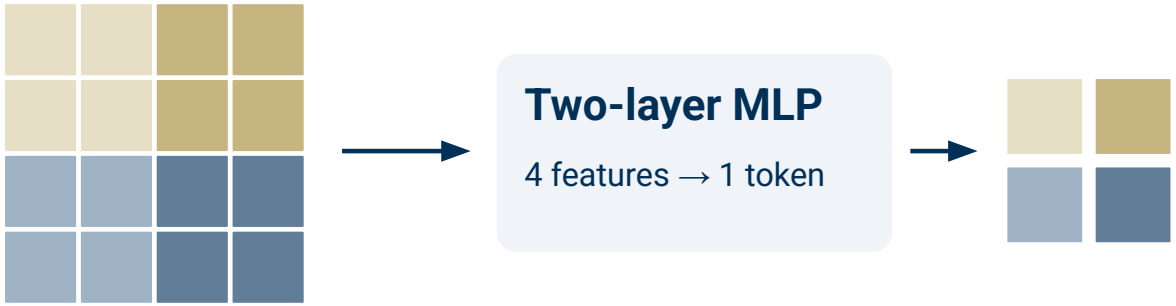
Starting point. Continue training a SigLIP-2 encoder at dynamic input resolutions; use 2D-RoPE plus interpolated absolute position embeddings.

Encoder sizes. SO-400M is the default; the 2B and 4B models use SigLIP2-Large (300M).

TENSOR SHAPES (SCHEMATIC)

$$F \in \mathbb{R}^{H_g \times W_g \times d_v} \longrightarrow V \in \mathbb{R}^{(H_g W_g/4) \times d_L}$$

The 2 × 2 MLP merger



$$\mathbf{v}_{r,c} = M\left(\left[\mathbf{f}_{2r,2c}; \mathbf{f}_{2r,2c+1}; \mathbf{f}_{2r+1,2c}; \mathbf{f}_{2r+1,2c+1}\right]\right)$$

Example: 64 × 64 features → 32 × 32 visual tokens
4,096 → 1,024 tokens, with the same LLM hidden width.

Fourfold visual-token reduction can cut the visual-only dense-attention score count by 16×; total runtime savings differ.

Report §2, p. 3 • formulas and grid are teaching schematics

Georgia Tech

Why one final visual representation is a bottleneck

A region-level semantic summary may be useful without preserving every detail needed for a later question.

A tiny but decisive detail

TOTAL

$3.87

Question: “What is the exact price?”

“A price on a receipt” identifies the region—but does not answer the question.

Final-layer-only transfer

Intermediate features

Potentially useful visual detail

↓

Final ViT features

One learned representation

The LLM must rely on whatever information survives this final interface.

DeepStack changes the route

Use multiple depths. Expose selected intermediate features as well as the main visual representation.

Inject them inside the LLM. The language model need not recover every relevant detail from a single input representation.

Not an explicit edge detector: “low-to-high level” is an intuition, not a guarantee about each layer.

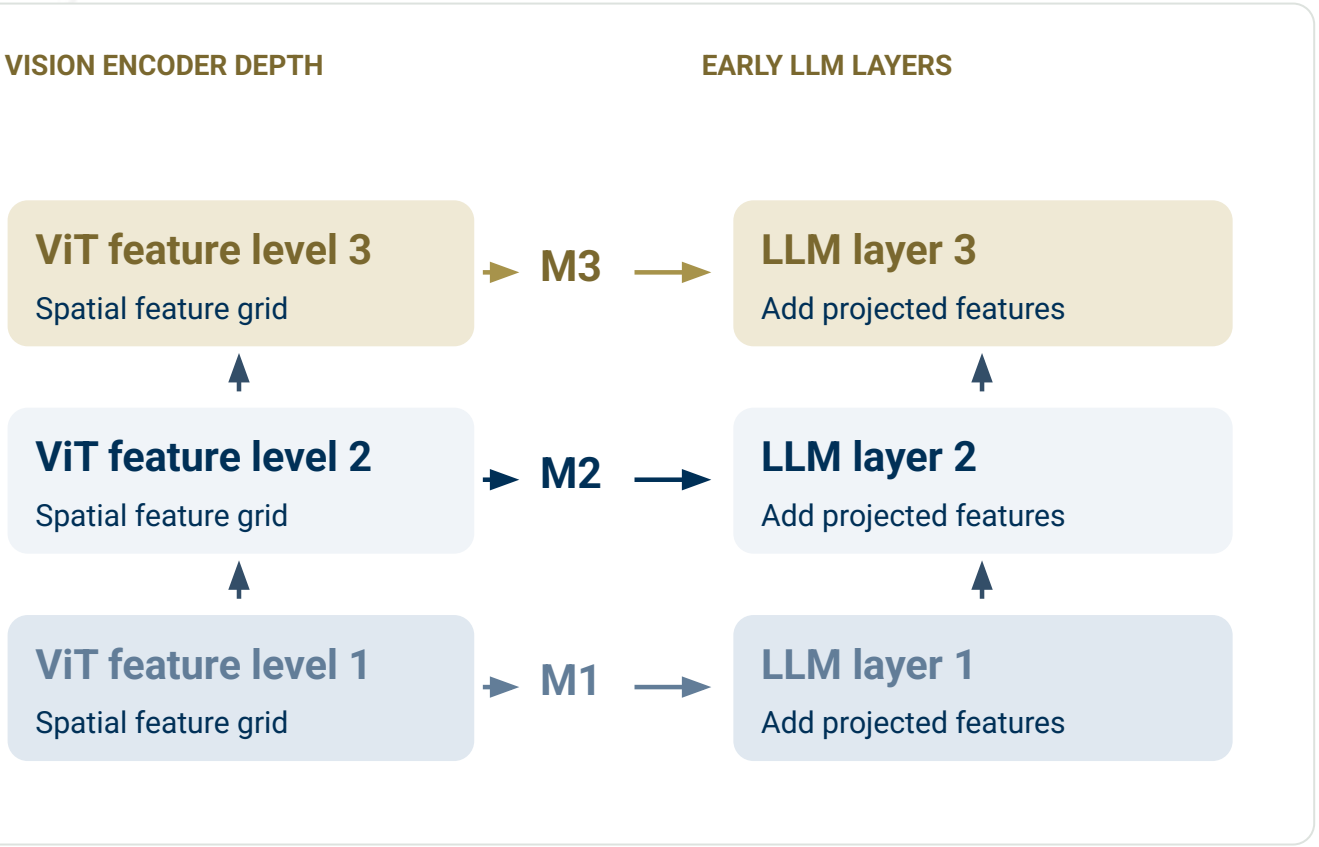
DeepStack does not undo compression; it gives the decoder additional learned routes to multi-level visual information.

Report §2.2, p. 4; §5.12.2, p. 24 • receipt is illustrative

Georgia Tech

Three feature sets enter the first three LLM layers

The important dimension is depth: add information to existing visual positions instead of appending more positions.



Aligned residual injection

$$D_k = M_k\left(F^{(a_k)}\right) \in \mathbb{R}^{N_v \times d_L}, \qquad k = 1, 2, 3$$

$$H^{(k)} = T_k\left(H^{(k-1)}\right) + S_{\mathcal{V}}(D_k)$$

Meaning. M□ maps a selected ViT feature set to LLM width; S_v places it at the corresponding visual positions.

Cost. Extra feature extraction, mergers, and additions—but no additional context positions from stacking.

Keep the distinction: multi-level features ≠ more image scales; unchanged context length ≠ zero extra computation.

Report §2.2 and Fig. 1, pp. 3–4 • equations are schematic

Georgia Tech

RoPE: encode position by rotating queries and keys

Teaching derivation: the key benefit is a relative-position term inside the attention score.

One two-dimensional feature pair

$$R(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}$$

$$\tilde{q}_p = R(p\omega_j)q_p, \qquad \tilde{k}_s = R(s\omega_j)k_s$$

For frequency ω_j, position p sets rotation angle $p\omega_j$. Apply the rotation to a query/key coordinate pair—not to an image itself.

Why relative offsets appear

$$\tilde{q}_p^{\top}\tilde{k}_s = q_p^{\top}R((s-p)\omega_j)k_s$$

Because $R(a)^{\top}R(b) = R(b-a)$, the positional part depends on $s - p$. Content vectors still matter; the entire score is not position-only.

Higher frequencies

Phase changes rapidly as position changes.

Lower frequencies

Phase changes slowly over larger offsets.

Different rotary frequencies provide different positional scales; Qwen3-VL changes how those scales are assigned to axes.

Background derivation; report §2.1 motivates its multimodal use, p. 3

Georgia Tech

Give time and both spatial axes a broad frequency range

The old issue was the allocation of frequencies—not an absence of spatial or temporal coordinates.

Blocked versus interleaved frequency assignment

t = temporal coordinate; h and w = the two spatial coordinates.

The mathematical idea

$$\theta_j = \omega_j \, p_{a(j)}, \qquad a(j) \in \{t, h, w\}$$

Before. Assigning axes to contiguous rotary dimensions can give each axis a different frequency band.

After. Interleave axes across the spectrum so each receives both low- and high-frequency components.

The report associates the more balanced spectrum with better long-video positional modeling; this is not a new attention operator.

Report §§1, 2.1, pp. 2–3 • layouts below are illustrative, not exact dimension counts

Georgia Tech

Separate temporal position from readable elapsed time

Qwen3-VL prefixes each video temporal patch with a timestamp string, while retaining multimodal positional encoding.

Time-linked ID limitations

Large, sparse indices. Tying temporal position IDs to absolute time can produce extreme index ranges in long videos.

Sampling burden. The report says effective learning required broad, uniformly distributed frame-rate sampling, increasing data-construction cost.

The issue is not simply “video frames lack timestamps.” It is how real time is represented.

Write time into the sequence

<3.0 seconds>

Visual group A

Spatial visual tokens follow the timecode.

→

<7.5 seconds>

Visual group B

Spatial visual tokens follow the timecode.

Now ask. “How long after A does B occur?” The sequence exposes 7.5 and 3.0 as readable time values; their difference is 4.5 seconds.

Training includes both seconds and hours:minutes:seconds formats. Text timestamps use a modest number of extra tokens.

MRoPE supplies positional structure; timestamp text makes actual time explicit. Neither guarantees perfect temporal reasoning.

Report §2.3, p. 4 • sequence and question are illustrative

Georgia Tech

Align first, train jointly, then extend the context

The four stages progressively increase both the trainable system and the scale of its sequential inputs.

Stage	Objective	Trainable parameters	Token budget	Sequence length
S0	Vision–language alignment	Merger only	67B	8,192
S1	Multimodal pretraining	All components	≈ 1T	8,192
S2	Long-context pretraining	All components	≈ 1T	32,768
S3	Ultra-long adaptation	All components	100B	262,144

Why start with merger-only training?

Bridge the feature-space mismatch while the pretrained encoder and language backbone remain fixed.

A long context is a shared token budget

$$S = N_{\text{text}} + N_{\text{visual}} + N_{\text{timestamp}} \leq 262{,}144$$

S2 expands long text, video, and agent data; S3 emphasizes long documents and long videos—not context length alone.

Report Table 1 and §3.1, pp. 4–5; long-document/video data §§3.2.3, 3.2.7

Georgia Tech

Balance modalities—and distinguish input length from loss weight

Long visual inputs, short answers, and long text responses can contribute very differently to optimization.

Keep learning from text

During pretraining. Text-only data remains in the mixture while vision, merger, and LLM parameters are trained jointly.

During post-training. Text data also supports instruction tuning and strong-to-weak distillation.

Goal ≠ universal outcome: the flagship does not beat its text-only counterpart on every language benchmark.

Two useful loss-normalization endpoints

EQUAL WEIGHT PER EXAMPLE

$$L_{\text{sample}} = \frac{1}{B}\sum_{i=1}^{B}\frac{1}{m_i}\sum_{t=1}^{m_i}\ell_{i,t}$$

Each sample has equal aggregate coefficient, regardless of how many tokens carry a loss.

EQUAL WEIGHT PER TOKEN

$$L_{\text{token}} = \frac{\sum_i \sum_{t=1}^{m_i} \ell_{i,t}}{\sum_i m_i}$$

A sample's aggregate coefficient grows with the number of supervised tokens.

What Qwen3-VL reports

Square-root-normalized per-token loss. The exact formula and treatment of masked tokens are not specified in this report.

In these background formulas, m_i counts supervised tokens—not automatically all image, video, and prompt positions.

Report §1, p. 2; §3.1, pp. 4–5; Tables 5–6, pp. 21–22 • endpoint formulas are teaching background

Georgia Tech

Build richer supervision, not just more image–text pairs

The corpus is engineered to teach fine detail, document structure, long-range evidence, and temporal relationships.

Image captions + entity knowledge

Recaption web pairs with a specialized Qwen2.5-VL-32B model. Use original text as context; describe attributes, layout, and interactions. Deduplicate recaptioned text and use visual-embedding clusters to identify sparse coverage.

OCR + layout-aware parsing

Combine OCR specialists and Qwen2.5-VL for pseudo-label refinement. Parse document regions in reading order, then reassemble position-aware content. Support QwenVL-HTML and QwenVL-Markdown annotations.

Long documents + interleaved books

Preserve page order and align text with figures. Build sequences up to 256K tokens and question–answer examples whose evidence spans pages, charts, tables, figures, and body text.

Temporally grounded video

Synthesize short-to-long, timestamp-interleaved captions; annotate objects and actions across time. Balance video sources and adjust frame sampling to the available sequence budget.

Better labels change what the network is asked to learn: attributes, relationships, layout, and evidence across time—not only category names.

Report §§3.2.1–3.2.3, pp. 5–6; §3.2.7, p. 8

Georgia Tech

Go beyond “what is it?” to “where—and what can I do?”

Grounding, spatial reasoning, multimodal code, and GUI trajectories make the output actionable.

2D grounding + counting

box: (x_1, y_1, x_2, y_2)
point: (x, y)

Locate. Train bounding-box and point responses for specific objects and arbitrary regions.

Normalize. Scale coordinates to [0, 1000] to make the output convention less dependent on raw resolution.

Counting: direct answers, boxes, and points.

Spatial relations + 3D

9-DoF box = position + size + roll / pitch / yaw

Relate. Learn relative positions, affordances such as “graspable,” and action-conditioned questions.

Ground in 3D. Predict structured boxes from a single-view image; unify labels in a virtual camera frame.

Monocular inference is learned—not guaranteed metric reconstruction.

Code + GUI behavior

screenshot → HTML / CSS
visual target → tool action

Generate. Convert visual material into HTML/CSS, SVG, code, and mathematical markup.

Interact. Train interface perception, grounding, multi-step GUI trajectories, and multimodal function calls.

Some function-call training responses are synthesized, not executed.

An output representation is part of the task design: words describe; coordinates localize; code and tool calls specify actions.

Report §§3.2.4–3.2.6, pp. 6–8; §3.2.9, p. 9; Appendix B.5, p. 37

Georgia Tech

Train the model to see correctly—and force the image to matter

A visual-math error can arise from misreading the diagram, faulty reasoning, or both.

Develop perception and reasoning

Perception supervision. Programmatically render diagrams; generate 1M point-grounding samples, 2M perception QA pairs, and 6M rich diagram captions.

Reasoning supervision. Curate over 60M educational exercises and over 12M image-paired long-reasoning samples; validate answers and filter trajectories.

Perception: "Which lines intersect?"
Reasoning: "What follows from that intersection?"

Multimodal-necessity filtering

Give only the text to Qwen3-30B-nothink

Test visual-mathematics training questions.

↓

Can it solve the problem correctly?

YES → discard

Text alone is enough for this model.

NO → retain for curation

A candidate requiring visual evidence.

This is an empirical filter—not proof of nonzero conditional mutual information.

High visual-benchmark accuracy is less meaningful when a model can ignore the visual input and exploit textual shortcuts.

Report §3.2.8, pp. 8–9; §4.2.2, p. 11

Georgia Tech

SFT produces two behaviors: Instruct and Thinking

The report separates output behavior through data and post-training—not through a new vision architecture.

Pretrained Qwen3-VL
General multimodal capability

→

Supervised fine-tuning
Instruction–response examples

→

Distillation → RL
Refine competence and behavior

Instruct / non-thinking

Target behavior. Useful direct answers and instruction following, without the explicit long-CoT format used for Thinking training.

SFT corpus. About 1.2M examples: one-third text-only, two-thirds image/video–text; single- and multi-turn conversations.

Thinking

Target behavior. Longer explicit reasoning trajectories, emphasizing difficult STEM and agentic tasks.

Cold-start data. Approximately 1:1 vision-language and text-only examples, with difficulty curation and response-quality filtering.

SFT itself is staged: begin at 32K, then train at 256K with long documents/videos mixed with shorter-context data.

Report §§4.1–4.2, pp. 9–11

Georgia Tech

Teacher demonstrations first; student trajectories next

The report uses text-only distillation to strengthen lightweight models' language backbone and multimodal performance.

1. Off-policy: learn the teacher's responses

Prompt x

A text-only training input

→

Strong teacher

Generate response y^T

↓

Student learns from demonstrations

Acquire a strong initial response policy.

"Off-policy" means the training responses were produced by the teacher, not by the current student.

2. On-policy: improve where the student goes

Student generates y^S

Use its own responses for the next phase.

$$p_T(\cdot \mid x, y^S_{<t}) \quad \longleftrightarrow \quad p_S(\cdot \mid x, y^S_{<t})$$

On each student-generated prefix, align the teacher's and student's next-token distributions using a KL-divergence objective.

The report does not specify the KL direction or full objective here.

Reasoning improvements can transfer to multimodal tasks, but text-only distillation is not a replacement for visual perception training.

Report §4.1, p. 9; §4.3, p. 11 • probability notation is schematic

Georgia Tech

Train on verifiable outcomes near the model's learning frontier

Qwen3-VL combines reasoning RL with general RL for instruction following, alignment, and behavioral correction.

Reasoning RL: objective feedback

$$\max_{\theta} \; \mathbb{E}_{y \sim \pi_\theta(\cdot|x)}[R(x, y)]$$

Tasks. Mathematics, code, logical reasoning, visual grounding, and puzzles with rule- or executor-based verification.

Training method. The report uses SAPO, a smooth adaptive policy-gradient method; it does not detail its update equations here.

General RL adds instruction-following, preference, language-consistency, and targeted error correction, using rules and model judges.

Curate problems that can teach

Remove initially unproductive queries. Sample 16 responses with a strong checkpoint; discard queries with no correct responses.

Then remove easy queries per model. Sample 16 responses and filter pass rates above 90%. The curated pool is about 30K queries.

$$\Pr(\text{no success in 16 trials}) = (1 - p)^{16}$$

Independent-trial illustration: zero observed successes does not mean a task is impossible.

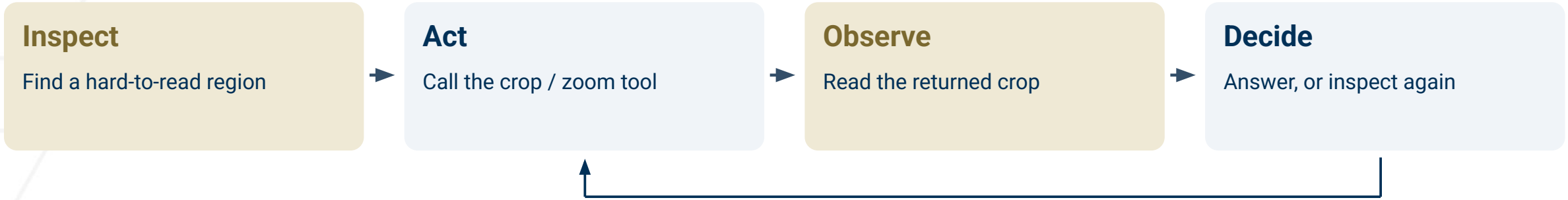
Reward design and data selection determine the learning signal. A verifier or judge is not automatically immune to reward hacking.

Report §4.4, pp. 11–12 • objective and sampling probability are explanatory

Georgia Tech

A visual agent can request new evidence before answering

Longer text reasoning cannot recover visual details that the model never inspected adequately.



Bootstrap the behavior

≈10K grounding examples
Train a Qwen2.5-VL-32B visual agent with SFT and tool-integrated RL.

≈120K multi-turn interactions
Distill and expand that behavior for Qwen3-VL post-training.

Reward the interaction—not only the final answer

Three rewards. Final-answer accuracy; correct interpretation of tool feedback; appropriate number of tool calls.

Observed failure. Models collapsed toward one tool call. An expert-estimated call-count target was added to discourage this shortcut.

“Thinking with images” is an observe–act–observe loop. Appropriate tool use is the goal—not more calls for their own sake.

Report §4.5, pp. 12–13; Appendix B.9, pp. 40–41 • crop example is illustrative

Georgia Tech

Thinking helps some tasks—not every task

Selected within-family comparisons for Qwen3-VL-235B-A22B, as reported in Table 2.

Benchmark	Instruct	Thinking	What the row tests
MathVision	66.5	**74.6**	Visual mathematical reasoning
MUIRBENCH	73.0	**80.1**	Multi-image understanding
DocVQA test	**97.1**	96.5	Document question answering
OCRBench	**920**	875	Text recognition; own score scale
Video-MME, no subtitles	**79.2**	79.0	General video understanding

Reasoning ≠ extraction

More reasoning does not guarantee better visual reading.

System outcomes, not component ablations

Compare within each benchmark; score scales differ.

A 256K context, a larger model, or a Thinking label is a capability setting—not a guarantee of correctness on every input.

Report Table 2, p. 15; §§5.2–5.4, pp. 14–17

Georgia Tech

Separate component ablations from whole-system claims

The report offers concrete controlled evidence for DeepStack—and narrower evidence for extreme context length.

DeepStack: matched pretraining comparison

Internal 15B-A2B model; 200B pretraining tokens; validation evaluation with no post-training.

Metric	Baseline	DeepStack
Reported average	74.7	**76.0**
InfoVQA	71.9	**74.2**
DocVQA	89.5	**91.1**
MMMU	52.9	**54.1**

The vision-encoder ablation also uses a matched 1.7B language backbone and 1.5T-token training.

Long context: retrieval ≠ general reasoning

VIDEO NEEDLE-IN-A-HAYSTACK

100%

Up to 256K tokens / 30 min

99.5%

≈1M tokens with YaRN extension

Scope. Locate an inserted salient frame and answer its question. This does not establish equally strong long-video reasoning.

Comparison caveat. Video frame caps differed: Qwen 2,048; Gemini 512; GPT-5 256; Claude 100. The authors flag imperfect fairness.

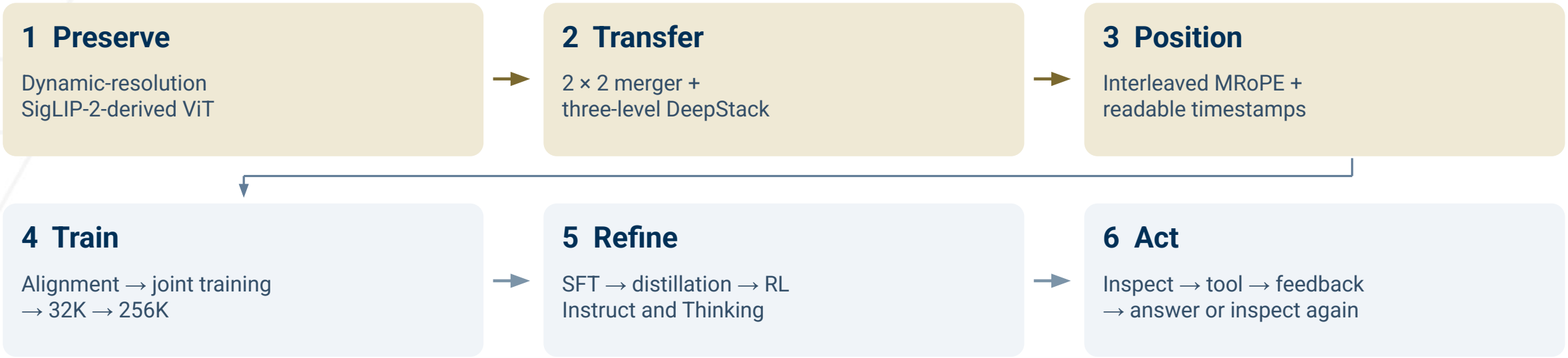
1M tokens is an extrapolation experiment—not the native 256K training window. Headline benchmark gains do not identify a single cause.

Report Tables 11–12, p. 24; Fig. 3 and §5.12.3, p. 25; §5.9, p. 20

Georgia Tech

The complete Qwen3-VL mental model

Trace the information: preserve it, compress it, position it, reason over it, and gather more when necessary.



THE THREE NEW ARCHITECTURAL IDEAS

DeepStack: multiple feature depths, not more sequence positions.
Interleaved MRoPE: balanced axis–frequency allocation.
Text timestamps: actual video time expressed in readable tokens.

The lesson is a system lesson: architecture, data, optimization, and tool behavior solve different parts of the multimodal problem.

Report §§2–4; interpretation grounded in the architecture, training recipe, and evaluations

Georgia Tech

References

Haotian Liu, Chunyuan Li, Qingyang Wu, Yong Jae Lee
Visual Instruction Tuning (2023)
arXiv:2304.08485v2

Qwen Team
Qwen3-VL Technical Report (2025)
arXiv:2511.21631v2

Georgia Tech