VLM: From Late Fusion to Early Fusion

VLM Fusion Strategies — Late Fusion vs. Early Fusion (Natively Multimodal)

Presenter: Hung-Chun Hsu

CFDA, CLIP & LKT Labs Joint Laboratory Seminar
Research Center for Information Technology Innovation
Academia Sinica, Taipei, Taiwan

May 22, 2026

About Me

- **Name**: Hung-Chun Hsu
- **Title**: Research Assistant, CFDA Lab
- **Education**: M.S. in Data Science, National Taiwan University
- **Research Interests**: GNN, LLM, Multimodal-LLM, Information Retrieval
- **Interests**: Guitar, Physics, Pure Mathematics, exploring new things

Follow-up — Gated DeltaNet v2 Released

Follow-up from last seminar: GDN v2 was released within 24 hours of our talk!



Ali Hatamizadeh • 2nd
LLM Tech Lead & Staff Research Scientist at NVIDIA
3h •

Gated DeltaNet-2 is here.

New paper: Gated DeltaNet-2: Decoupling Erase and Write in ...more

NVIDIA 2026-5-21

Gated DeltaNet-2: Decoupling Erase and Write in Linear Attention

Ali Hatamizadeh Yejin Choi Jan Kautz
{ahatamizadeh, yejinc, jkautz}@nvidia.com

Abstract: Linear attention replaces the unbounded cache of softmax attention with a fixed-size recurrent state, reducing sequence mixing to linear time and decoding to constant memory. The hard part is not just what to forget, but how to edit this compressed memory without scrambling existing associations. Delta-rule models subtract the current read before writing a new value, and Kimi Delta Attention (KDA) sharpens forgetting with channel-wise decay. But the active edit still uses a single scalar gate to control two different things, how much old content to erase on the key side and how much new content to commit on the value side. We introduce Gated DeltaNet-2, which generalizes both Gated DeltaNet and KDA by inheriting adaptive forgetting and channel-wise decay while addressing their shared limitation, the scalar tie between erasing and writing. Gated Delta Rule-2 separates these roles with a channel-wise erase gate b_t and a channel-wise write gate w_t, reducing to KDA when both gates collapse to the same scalar and to Gated DeltaNet when the decay also collapses. We derive a fast-weight update view, a chunkwise WY algorithm with channel-wise decay absorbed into asymmetric erase factors, and a gate-aware backward pass that preserves efficient parallel training. At 1.3B parameters trained on 100B FineWeb-Edu tokens, Gated DeltaNet-2 achieves the strongest overall results among Mamba-2, Gated DeltaNet, KDA, and Mamba-3 variants across language modeling, commonsense reasoning, and retrieval. Its advantage is most pronounced on long-context RULER needle-in-a-haystack benchmarks, where it improves the evaluated multi-key retrieval setting and remains strong in both recurrent and hybrid settings.
Code: https://github.com/NVlabs/GatedDeltaNet-2

https://github.com/NVlabs/GatedDeltaNet-2

Chapter Overview

1. **Late Fusion: Qwen3-VL as a Case Study**
   - DeepStack architecture and MLP Merger selective injection
   - 7-stage training pipeline (Pre-training S0–S3 + Post-training PT-1–PT-3)
   - Limitations of late fusion: ViT Tax and the small-model bottleneck
   - Evolution of the Qwen image encoder family
2. **Early Fusion: Qwen3.5 as a Case Study**
   - Architecture overview (unified backbone + GDN/GA hybrid attention)
   - 7-stage training pipeline (Pre-training S0–S2 + Post-training PT-1–PT-4)
   - No more separate "-VL" models starting from Qwen3.5
3. **Late Fusion vs. Early Fusion Comparison**
4. **Related Tools** — veRL framework and VeOmni
5. **Summary and Future Outlook**

References and Acknowledgements

Reference Papers:

Source	Title	Authors
arXiv:2511.21631	Qwen3-VL Technical Report	Qwen Team
Blog, 2026	Qwen3.5: Towards Native Multimodal Agents	Alibaba Qwen Team
arXiv:2604.11554	Relax: Async RL Engine for Omni-Modal	Xiaohongshu AI
ICLR 2025	Gated DeltaNet	Yang, Kautz, Hatamizadeh

Acknowledgements:

Qwen Official Blog
Alibaba Cloud Blog
TrilogyAI Deep Dive: Qwen3.5
The Salt: Qwen3-VL Architecture

2026 Native Multimodal Model Trend

Which major language models adopted Early Fusion (natively multimodal) in 2026?

Open-Source Models:

Model	Developer	Released	Architecture Highlights
Llama 4	Meta	2025-04	Among the first open-source early-fusion models
Qwen3.5	Alibaba	2026-02	Early Fusion; GDN + MoE hybrid
DeepSeek V4	DeepSeek	2026-04	Early Fusion

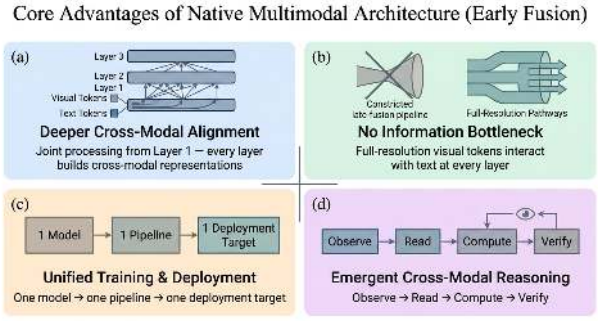
Closed-Source Models:

Model	Developer	Released	Architecture Highlights
GPT-5	OpenAI	2025-08	Early Fusion
GPT-5.5	OpenAI	2026-04	Early Fusion
Gemini 3.5 Flash	Google	2026-05	Early Fusion

Still using Late Fusion: Qwen3-VL, Gemma 3n, Mistral Small 4, Kimi-VL

Core Advantages of Early Fusion (Natively Multimodal)

Why is the industry shifting to Early Fusion? Four core advantages:



1. **Deeper Cross-Modal Alignment** — Vision and text tokens jointly processed from Layer 1.
2. **Elimination of the Information Bottleneck** — No single projection layer compressing all visual information.
3. **Unified Training and Deployment** — One model, one pipeline, one deployment target.
4. **Emergent Cross-Modal Reasoning** — Dynamic re-focus on image regions during reasoning.

Qwen3-VL vs. Qwen3.5 — Quick Comparison

Aspect	Qwen3-VL	Qwen3.5
Released	2025-09	2026-02
Fusion Strategy	Late Fusion	**Early Fusion**
Architecture	ViT + Patch Merger + Transformer LLM	Unified GDN + Attention hybrid backbone
Vision Processing	ViT processes independently, then projects into LLM	Vision tokens jointly processed with text from Layer 1
Model Family	Separate: Qwen3 (text) + Qwen3-VL (VL)	Single family; every model is natively multimodal
Inference Speed	Baseline (Qwen3-Max)	32K: **8.6**×↑, 256K: **19**×↑
MMMU	80.6	**85.0** (+4.4)
ERQA	52.5	**67.5** (+15.0)
BabyVision	22.2	**52.3** (+30.1)

1. Late Fusion: Qwen3-VL Case Study

▶ **1. Late Fusion: Qwen3-VL Case Study**

2. Early Fusion: Qwen3.5 Case Study

3. Late Fusion vs. Early Fusion Comparison

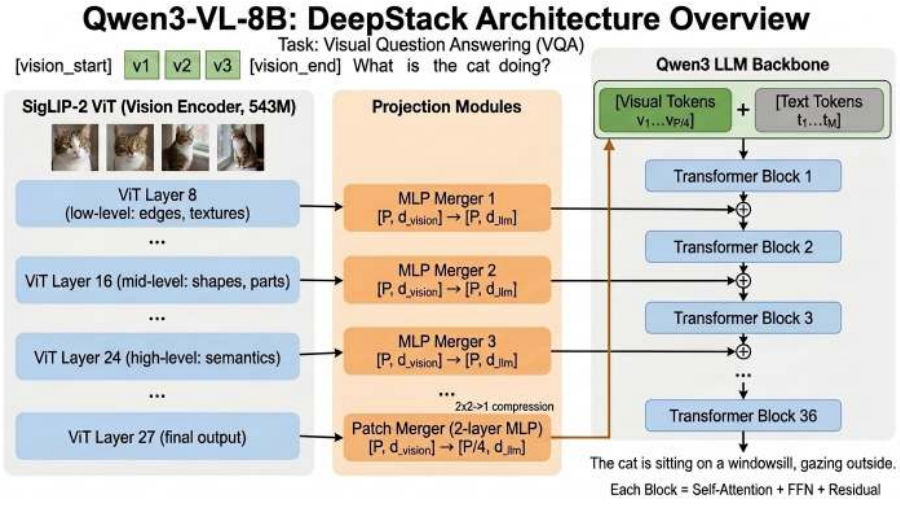
4. Related Tools: Qwen3.5 Natively Multimodal

5. Summary & Future Outlook

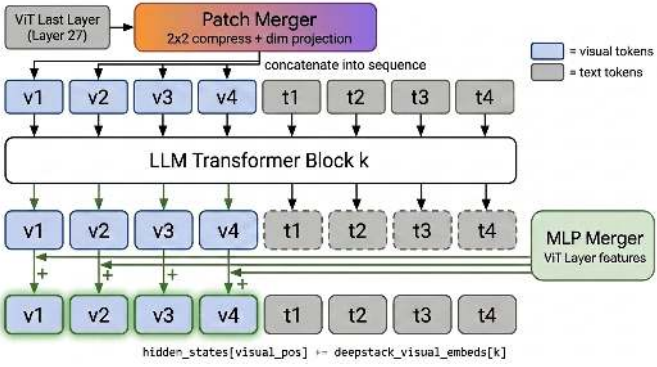
1.1 Qwen3-VL Three-Component Architecture

1. Late Fusion: Qwen3-VL Case Study
   - ▶ **1.1 Qwen3-VL Three-Component Architecture**
   - 1.2 DeepStack Multi-Layer Visual Feature Injection
   - 1.3 Pre-Training Four Stages
   - 1.4 Limitations of Late Fusion

Qwen3-VL DeepStack Architecture Overview



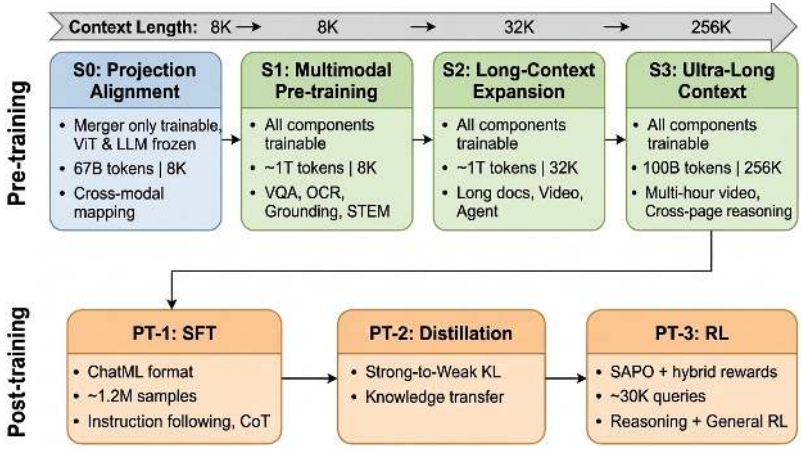
DeepStack MLP Merger Selective Injection



Three-layer injection paths:

	Path 1	Path 2	Path 3
ViT Layer	8	16	24
Feature Level	Low (edges, textures)	Mid (shapes, parts)	High (semantics)
Projector	MLP Merger 1	MLP Merger 2	MLP Merger 3
LLM Target	After Block 1	After Block 2	After Block 3

Seven-Stage Training Pipeline



Qwen3-VL (arXiv:2511.21631): **7-stage progressive pipeline** (4 pre-training + 3 post-training) — freeze all backbones first, then gradually unfreeze and expand.

Core idea: let the Merger bridge the modality gap first, then jointly optimize all components.

Pre-Training Four Stages (S0–S3)

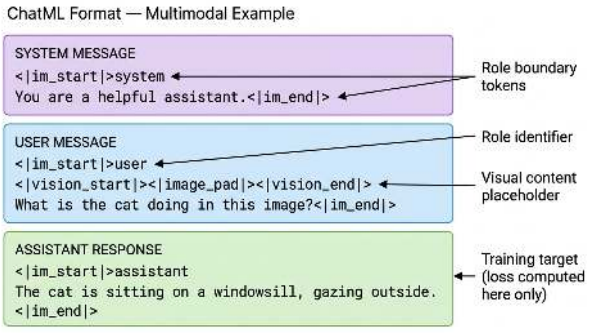
Stage	Objective	Trainable	Tasks	Scale	Seq Len
S0	Vision-language projection alignment	Merger only	Image-text pairs, OCR	67B	8K
S1	Diverse visual understanding	ViT+Merger+LLM	Captioning, VQA, grounding, OCR, STEM	∼1T	8K
S2	Long-context capability	ViT+Merger+LLM	Long documents, video, agent instructions	∼1T	32K
S3	Ultra-long context adaptation	ViT+Merger+LLM	Long video, long document reasoning	100B	256K

- **S0**: ViT and LLM both **frozen** — only Merger trained to learn cross-modal mapping in fixed visual and language spaces
- **S1**: First full end-to-end training; largest data stage
- **S2**: Sequence length $4\times$ expansion ($8\text{K} \rightarrow 32\text{K}$); video and agent data ratio increases significantly
- **S3**: Sequence length $8\times$ expansion ($32\text{K} \rightarrow 256\text{K}$); focuses on scarce high-quality ultra-long data

More details: Qwen3-VL Technical Report (arXiv:2511.21631), §3.1 Training Recipe, §3.2 Pre-Training Data (§3.2.1–§3.2.9).

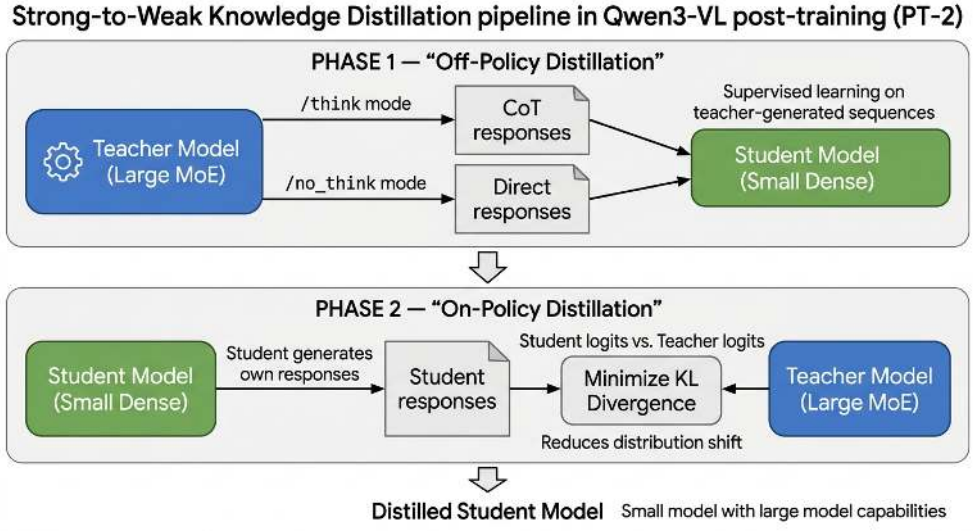
Post-Training Three Stages (PT-1–PT-3)

Stage	Objective	Method	Tasks	Scale
PT-1	Instruction following & CoT	SFT (ChatML format)	Multimodal dialogue, CoT, long doc/video QA	∼1.2M samples
PT-2	Large→small capability transfer	Strong-to-Weak Distillation	Teacher response distillation, text-only SFT	—
PT-3	Reasoning & general enhancement	SAPO + hybrid rewards	Math, code, VQA, OCR, visual grounding	∼30K queries



More details: Qwen3-VL Technical Report (arXiv:2511.21631), §4.1 Training Recipe, §4.3 Strong-to-Weak Distillation, §4.4 RL.

Supplement — Strong-to-Weak Distillation

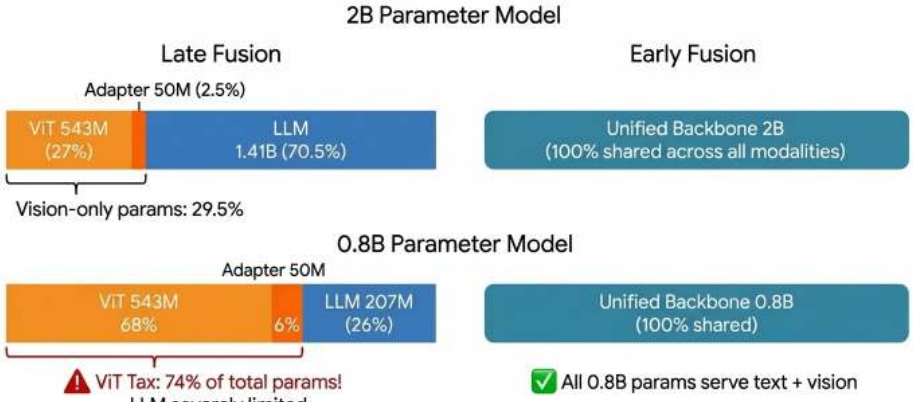


Teacher: **Qwen3-VL-235B-A22B** (MoE flagship) → Students: Dense 2B / 4B / 8B + MoE 30B-A3B

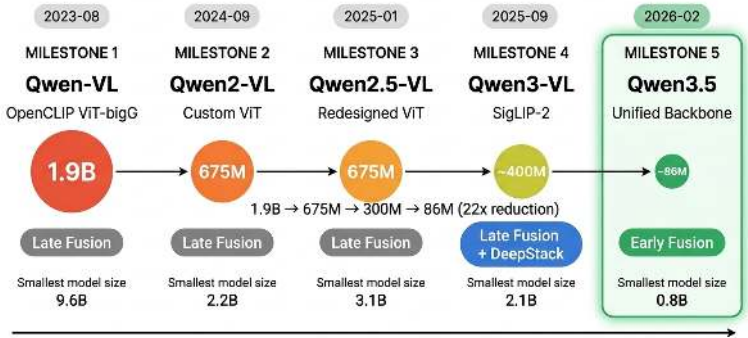
Late Fusion Limitation — Vision Encoder Overhead in Small Models

Problem: A fixed-size vision encoder dominates the parameter budget in small models.

```
Late Fusion 2B:  ViT (~400M, 27%) + Adapter + LLM (1.41B, 70%)
Late Fusion 0.8B: ViT (~400M, 68%) + Adapter + LLM (207M, 26%) ← LLM severely under-budgeted
Early Fusion 0.8B: Unified backbone (0.8B, 100% shared across modalities)
```



Qwen Image Encoder Evolution



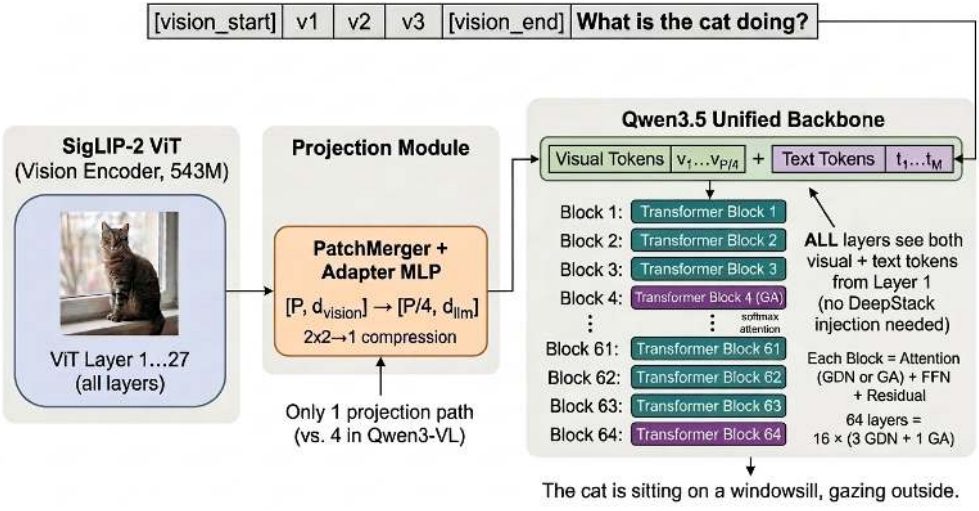
Generation	Release	Fusion	Image Encoder	Enc. Params	Smallest Model	Total Params	ViT Tax
Qwen-VL	2023-08	Late Fusion	OpenCLIP ViT-bigG	**1.9B**	Qwen-VL	∼9.6B	20%
Qwen2-VL	2024-09	Late Fusion	Custom ViT	**675M**	Qwen2-VL-2B	∼2.2B	31%
Qwen2.5-VL	2025-01	Late Fusion	Redesigned ViT	**675M**	Qwen2.5-VL-3B	∼3.1B	22%
Qwen3-VL	2025-09	Late + DeepStack	SigLIP2-Large†	**∼300M**	Qwen3-VL-2B	∼2.1B	14%
Qwen3.5	2026-02	**Early Fusion**	Unified backbone‡	**∼86M**	Qwen3.5-0.8B	**0.8B**	—

† Qwen3-VL 2B/4B uses SigLIP2-Large (∼300M); 8B+ uses SigLIP2-So400m (400M). ‡ Qwen3.5 vision frontend varies by model size: 0.8B uses ViT-B (∼86M), 4B uses ViT-L (∼304M), 9B/27B uses So400m (∼413M). Table shows smallest model.

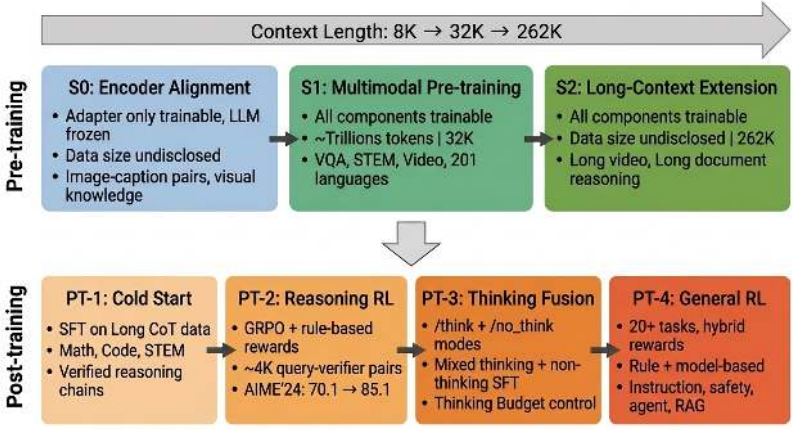
2. Early Fusion: Qwen3.5 Case Study

1. Late Fusion: Qwen3-VL Case Study
▶ **2. Early Fusion: Qwen3.5 Case Study**
3. Late Fusion vs. Early Fusion Comparison
4. Related Tools: Qwen3.5 Natively Multimodal
5. Summary & Future Outlook

Qwen3.5 Early Fusion Architecture Overview



Qwen3.5 Seven-Stage Training Pipeline



No More Separate "-VL" Models Since Qwen3.5

Generation	Release	Fusion	Text Model	VL Model	Two Models?
Qwen	2023-08	Late Fusion	Qwen	Qwen-VL	✓
Qwen2	2024-09	Late Fusion	Qwen2	Qwen2-VL	✓
Qwen2.5	2025-01	Late Fusion	Qwen2.5	Qwen2.5-VL	✓
Qwen3	2025-09	Late + DeepStack	Qwen3	Qwen3-VL	✓
Qwen3.5	2026-02	**Early Fusion**	—	—	**✗ Unified**

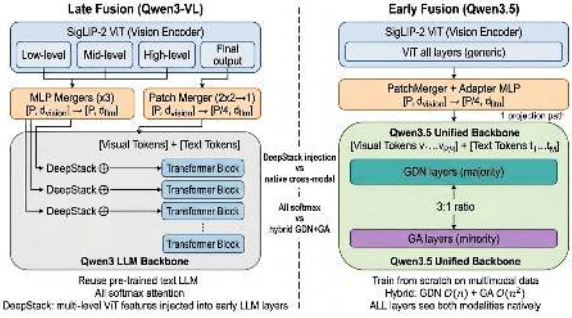
Qwen3.5 Model	Total Params	Active Params	Type
Qwen3.5-0.8B	0.8B	0.8B	Dense
Qwen3.5-2B / 4B / 9B	2–9B	2–9B	Dense
Qwen3.5-27B	27B	27B	Dense
Qwen3.5-35B-A3B	35B	3B	MoE
Qwen3.5-397B-A17B	397B	17B	MoE

Key insight: From Qwen3.5 onward, **every model is natively multimodal** — one set of weights handles text, images, and video. No more separate text vs. VL editions.

All Qwen3.5 models feature a 262K native context window, extendable to 1,010,000 tokens via YaRN RoPE scaling.

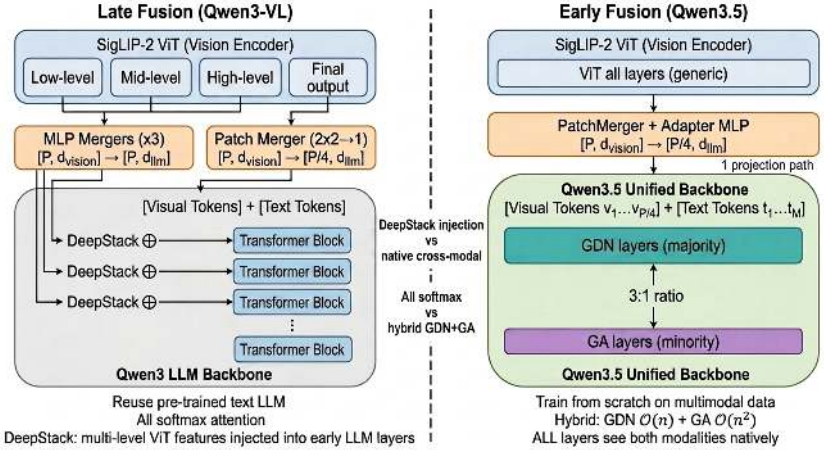
1. Late Fusion: Qwen3-VL Case Study
2. Early Fusion: Qwen3.5 Case Study
▶ **3. Late Fusion vs. Early Fusion Comparison**
4. Related Tools: Qwen3.5 Natively Multimodal
5. Summary & Future Outlook

Training Philosophy Differences



Aspect	Qwen3-VL (Late Fusion)	Qwen3.5 (Early Fusion)
LLM backbone origin	Reuses pretrained Qwen3	Trained from scratch (unified)
Modal token handling	ViT → projection → LLM input	Vision + text **same layer, same sequence**
Compute cost	Incremental (reuse text LLM)	Full (train unified backbone)
Modality scalability	New encoder + adapter per modality	Unified backbone supports all

Training Philosophy Differences — Cost vs. Payoff



Early Fusion — Training Cost:

- Cannot reuse existing text models
- Requires larger-scale multimodal corpora
- Higher stability demands (cross-modal gradient

Early Fusion — Training Payoff:

- Every layer understands both modalities
- One model replaces Qwen3 + Qwen3-VL
- RL gradients propagate directly to all modal

Late Fusion vs. Early Fusion — Architecture Comparison

Aspect	Qwen3-VL (Late Fusion)	Qwen3.5 (Early Fusion)
Fusion Strategy	Late Fusion	Early Fusion
Visual Token Integration	ViT processes independently → projection → injection at specific LLM layers	Visual tokens enter the shared backbone alongside text tokens from the earliest layer
Cross-Modal Representation	Starts only after visual features are injected into the LLM	Co-evolves at every layer, from bottom to top
Model Unity	Separate families (Qwen3 + Qwen3-VL)	Single unified family; every model is natively multimodal

Four Reasons Early Fusion Wins



1. Deeper Cross-Modal Alignment

- Every layer sees both modalities from Layer 1
- Late fusion: bottom layers never see visual info



2. No Information Bottleneck

- Full-resolution visual tokens at every layer
- Late fusion: all vision compressed through one projection



3. Unified Training & Deployment

- One model, one objective, one pipeline
- No separate text model + VL adapter



4. Emergent Cross-Modal Reasoning

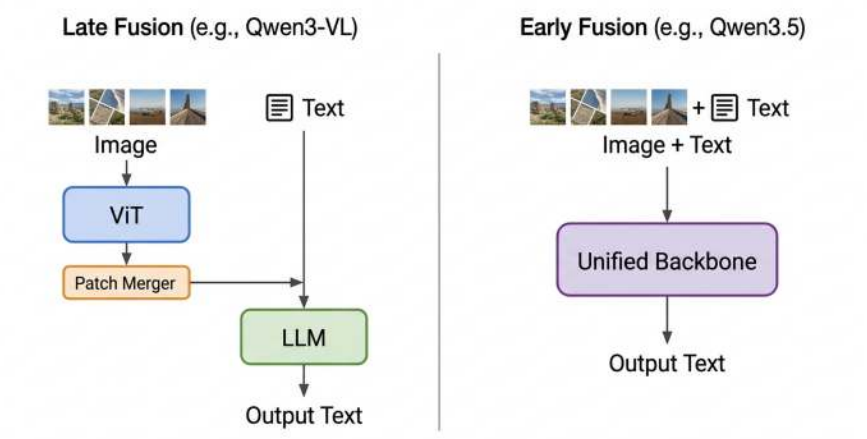
- Jointly reason over code + charts + text
- Hard when modalities processed separately

Training Complexity Overview — Three Levels

"Is early fusion training more expensive?" — It depends on which level you ask about.

Comparison Level	Late Fusion	Early Fusion	Higher?
① **Naive per-step attention complexity**	Lower (separate short sequences)	Higher (merged long sequence)	Early
② **Actual per-step throughput** (with modern techniques)	Loss due to ViT pipeline sync	∼100% (GDN + MoE)	**Late worse**
③ **Total training cost to reach equivalent capability**	Much lower (reuse pre-trained components)	Much higher (train unified backbone from scratch)	**Early sig. higher**

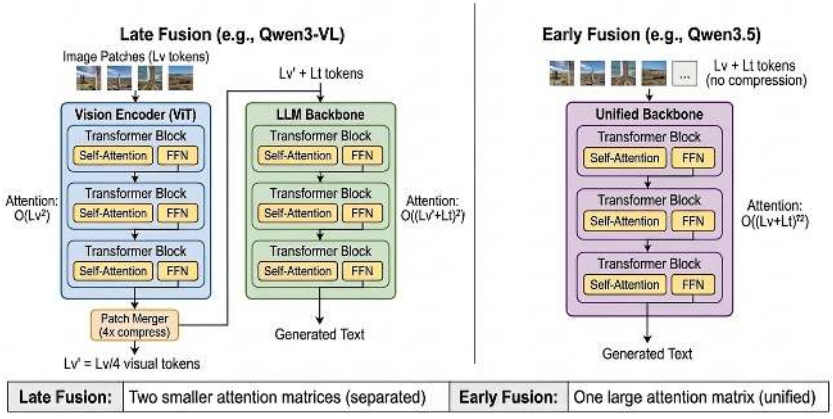
Per-Step Attention Complexity — Why Naive Early Fusion Costs More



Left: Separate vision encoder + language model

Right: Single unified model for all modalities

Per-Step Attention Complexity — Why Naive Early Fusion Costs More



ViT also has attention: Late Fusion's ViT has complexity $O(L_v^2)$. Combined with the LLM's $O\big((L_v'+L_t)^2\big)$, still less than Early Fusion's $O\big((L_v+L_t)^2\big)$ — separate computation eliminates the cross-modal term $2L_vL_t$, and Patch Merger compresses $L_v \to L_v'$.

Per-Step Attention Complexity — Math Analysis

Late Fusion

Two separate, smaller matrices

$$\underbrace{O(L_v^2)}_{\text{ViT}} + \underbrace{O\big((L_v' + L_t)^2\big)}_{\text{LLM}}$$

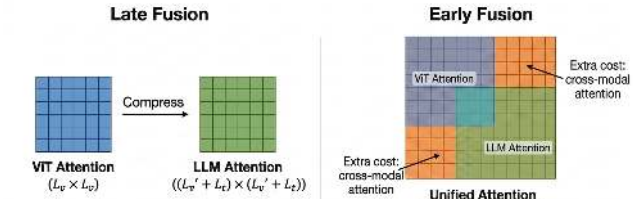
L_v (visual tokens)	784
L_v' (after Patch Merger)	256
L_t (text tokens)	512
ViT seq len	$L_v = 784$
LLM seq len	$L_v' + L_t = 768$
Largest matrix	$784^2 = 614$K

Early Fusion

One large matrix

$$O\big((L_v + L_t)^2\big) \times \text{all layers}$$

L_v (visual tokens)	784
L_t (text tokens)	512
Seq len	$L_v + L_t = 1{,}296$
Attention matrix	$1{,}296^2 = \mathbf{1{,}680}$K
vs. Late Fusion LLM	$\sim$**2.2**$\times$ **larger**



Benchmark Evidence — MMMU / ERQA / OmniDocBench

Qwen3-VL → Qwen3.5 vs. contemporary competitors:

Benchmark	Qwen3-VL 235B-A22B	Qwen3.5 35B-A3B	Qwen3.5 397B-A17B	GPT-5.2	Claude Opus 4.5
Release date	2025-09	2026-02	2026-02	2026-04	2026-04
Active params	22B	**3B**	17B	—	—
MMMU	80.6	>80.6	**85.0** (+4.4)	—	—
OmniDocBench v1.5	—	—	**90.8**	85.7	87.7
ERQA	52.5	—	**67.5** (+15.0)	—	—

Qwen3.5-35B-A3B surpasses Qwen3-VL-235B-A22B (22B active) on MMMU with only **3B active parameters** — a direct benefit of the early fusion architecture and hybrid GDN+GA design.

Fusion Strategy Trends 2025–2026

Open-Source Models (newest first):

Model	Developer	Release	Fusion Strategy
DeepSeek V4	DeepSeek	2026-04	**Early Fusion**
Mistral Small 4	Mistral AI	2026-03	Late Fusion
Qwen3.5	Alibaba	2026-02	**Early Fusion**
Qwen3-VL	Alibaba	2025-09	Late Fusion + DeepStack
Gemma 3n	Google	2025-06	Late Fusion
Llama 4	Meta	2025-04	**Early Fusion**
Phi-4-Multimodal	Microsoft	2025-02	Mixture-of-LoRAs
DeepSeek Janus-Pro	DeepSeek	2025-01	Decoupled Fusion

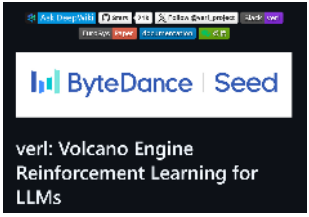
Closed-Source Models (newest first):

Model	Developer	Release	Fusion Strategy
Gemini 3.5 Flash	Google	2026-05	**Early Fusion**
GPT-5.5	OpenAI	2026-04	**Early Fusion**
Claude Opus 4.7	Anthropic	2026-04	Latent Embedding Fusion
GPT-5	OpenAI	2025-08	**Early Fusion**

4. Related Tools

1. Late Fusion: Qwen3-VL Case Study
2. Early Fusion: Qwen3.5 Case Study
3. Late Fusion vs. Early Fusion Comparison
▶ **4. Related Tools**
5. Summary & Outlook

veRL Framework & Early-Fusion Support



veRL (volcengine Reinforcement Learning):

The most widely used open-source multimodal RL training framework (ByteDance Seed team). Supports GRPO, PPO, and other algorithms.

Supported Multimodal Models (all Late-Fusion):

Model	Fusion Type	Support Version
Qwen2.5-VL	Late-Fusion (ViT + LLM)	Primary documented example
Qwen3-VL (Dense & MoE)	Late-Fusion (SigLIP2 + DeepStack)	Added in v0.6.1
Kimi-VL	Late-Fusion	Listed in README

Early-Fusion Support Status (Qwen3.5, Llama 4):

- veRL v0.7.1 added Qwen3.5 support, but **text-only LLM training only**
- Issue #5656: shape mismatch error when training Qwen3.5-A17B in multimodal mode
- **Omni-model RL training** planned for **v0.9 roadmap**

VeOmni — Omni-Modal Training Framework



VeOmni (ByteDance Seed):

A framework designed specifically for omni-modal training. Planned integration with veRL to support RL training for early-fusion models.

Current Status:

- ✓ Qwen3.5 **SFT / DPO training** supported (config available since v0.1.8+)
- ✗ **RL training (GRPO/PPO via veRL) not yet supported** (Issue #262 still open)
- ○ Integration expected in veRL v0.9, enabling RL post-training for unified-backbone models like Qwen3.5

https://github.com/ByteDance-Seed/VeOmni

5. Summary & Outlook

1. Late Fusion: Qwen3-VL Case Study
2. Early Fusion: Qwen3.5 Case Study
3. Late Fusion vs. Early Fusion Comparison
4. Related Tools
▶ **5. Summary & Outlook**

Summary & Takeaways

	Late Fusion (Qwen3-VL)	**Early Fusion (Qwen3.5)**
Architecture	ViT + Merger + LLM (3 components)	Unified backbone (single model)
Vision	ViT independent → inject into LLM	Joint from Layer 1
Cross-Modal	Starts at injection layers only	Every layer, bottom to top
Deployment	Two families (text + VL)	One model, all modalities
RL Gradients	Indirect (adapter bottleneck)	Direct, unified path

Takeaways:

1. **Fusion point determines the ceiling** — ERQA +15.0, BabyVision +30.1 vs. prior gen
2. **Unified architecture = unified pipeline** — no more two-model deployment
3. **Industry direction is set** — Llama 4, GPT-5, Qwen3.5: all early fusion
4. **Hybrid attention complements early fusion** — GDN $O(n)$ offsets 2.2× attention growth
5. **Open-source RL tooling lags** — veRL late-fusion-centric; VeOmni v0.9 is key

Supplementary

The following are supplementary technical materials for post-session reference.

Topic	Description
MRoPE	Enhanced Interleaved-MRoPE Visual Position Encoding
Relax	Async RL Engine — Three-Layer Architecture for Omni-Modal Post-Training

Supplementary — MRoPE Visual Position Encoding

Enhanced Interleaved-MRoPE (Multi-dimensional Rotary Position Embedding)

- Enhanced MRoPE assigns full-frequency encoding across three dimensions: **temporal**, **width**, and **height**
- Strengthens spatial modeling for images and temporal reasoning for videos
- **Text-Based Time Alignment**: evolved from T-RoPE to explicit text-timestamp alignment
  - More precise temporal localization and event grounding
  - Improved performance on long-duration video understanding tasks

Key idea: MRoPE enables the model to simultaneously maintain the spatial arrangement of image patches and the temporal ordering of video frames, while text-based time alignment further improves precise event localization in videos.

Supplementary — Relax Async RL Engine

Layer	Name	Function
1	Omni-Native Architecture	Multimodal support built end-to-end (data preprocessing, modality-aware parallel strategies, inference generation)
2	Independent Fault-Tolerant Service	Each RL role (actor, critic, reward model) operates independently; can scale, recover, and upgrade independently
3	Service-Level Decoupling	Async training via **TransferQueue** data bus; single staleness parameter controls on-policy $\leftrightarrow$ fully async

Configuration	Speedup	Notes
Qwen3-4B on-policy (vs. veRL)	**1.20**×	Baseline comparison
Qwen3-4B fully async	**1.76**×	Same model, async mode
Qwen3-Omni-30B fully async	**2.00**×	Larger model benefits more

Key result: Stable omni-modal RL convergence on image, text, and audio; video scenarios sustain $>$**2,000 steps** without degradation.

Relax (Xiaohongshu AI Platform, Apache 2.0): arXiv:2604.11554 | https://github.com/redai-infra/Relax

Thank You for Listening!

Questions & Discussion