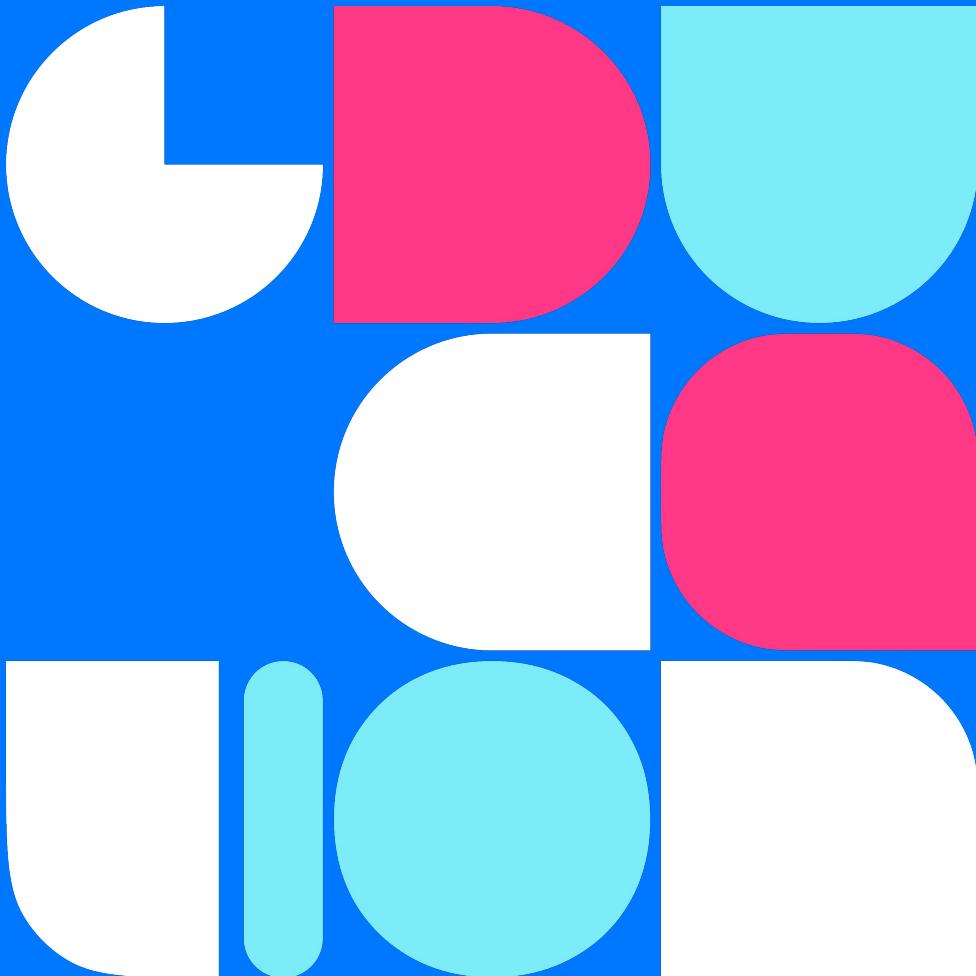




Продвинутые Техники

Курс "Мультимодальные модели"



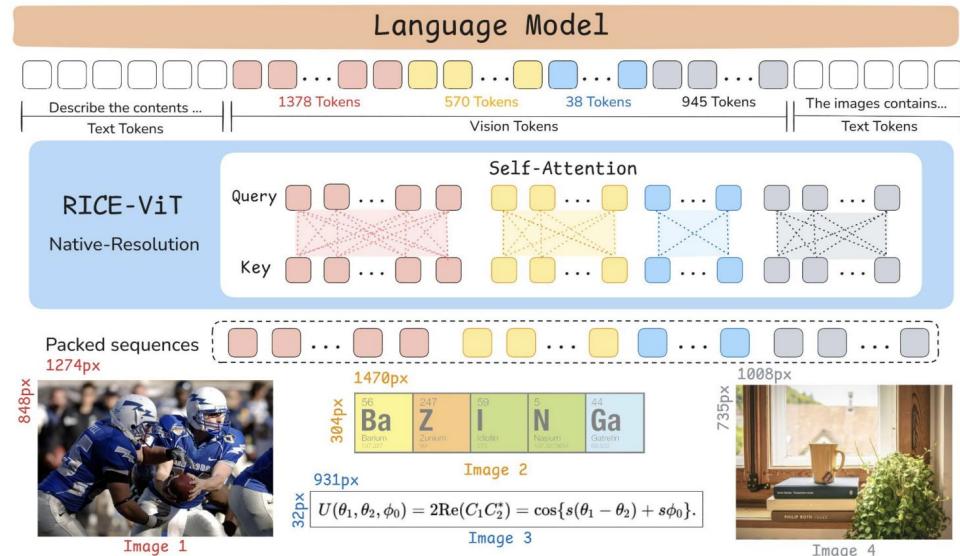
Цель занятия:

Провести обзор последних
улучшений и моделей в области VLM,
Omni моделей и посмотреть на
инсайты, связанные с обучением и
поведением моделей.

1. Общие улучшения в VLM

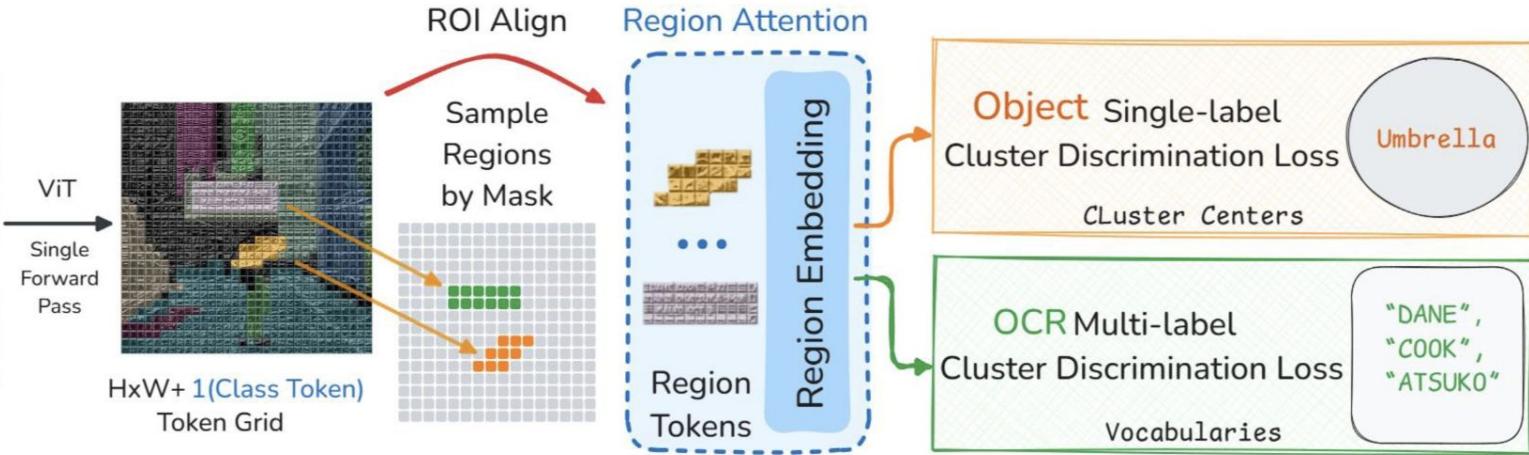
LLaVA-OneVision-1.5

- Основной вклад: SOTA-качество при минимальных ресурсах (обучение < \$16 000), целились в обгон QWEN-2.5-VL
- Улучшаются без сложных архитектур за счет новых энкодеров и LLM + данных.
- Vision Encoder: RICE-ViT (Xie et al., 2025)
 - обучен на 450 М изображениях, 2.4 В регионах.
 - улучшает region-level понимание и OCR.
 - использует 2D rotary positional encoding, масштабируется по разрешению.
- LLM Backbone: Qwen3.
- Projector: два слоя MLP, объединяет группы из 4 патчей → текстовое пространство LLM.



Overall architecture of LLaVA-OneVision-1.5. The framework integrates a pre-trained vision encoder with a language model decoder. The vision encoder adopts 2D RoPE for native-resolution processing and incorporates region-aware attention to enhance local semantic modeling. During pretraining, both object regions and OCR regions are jointly modeled to inject fine-grained text understanding capability. A lightweight projector maps visual features into the LLM embedding space, and the [CLS] token is preserved to retain global semantic capacity during multimodal alignment.

LLaVA-OV-1.5. RICE-ViT



Проблема CLIP/SigLIP

Классические модели (CLIP, SigLIP) обучаются на instance-wise contrastive loss: каждая картинка = один embedding.

Все примеры считаются независимыми (все "негативы"), поэтому теряется структура семантических сходств между регионами.

Такие модели плохо различают частично схожие объекты (например, "кошка в коробке" vs "кошка на столе").

Также страдают OCR и мелкие детали, текст внутри изображений не моделируется на уровне регионов.

Region-Aware Cluster Discrimination (RCD) Loss

Разбиваем изображение на семантические регионы с помощью ViT-attention и детектора регионов.

Вместо instance-wise используем кластеризацию регионов по смыслу. В процессе обучения регионы с похожим содержанием объединяются в кластеры (положительные пары); регионы из разных кластеров — отрицательные.

Пусть v_i - embedding региона, c_i - центр его кластера. Loss сводится к минимизации:

$$\mathcal{L}_{\text{RCD}} = - \sum_i \log \frac{\exp(\text{sim}(v_i, c_i)/\tau)}{\sum_j \exp(\text{sim}(v_i, c_j)/\tau)}$$

LLaVA-OV-1.5. Сравнение визуальных энкодеров

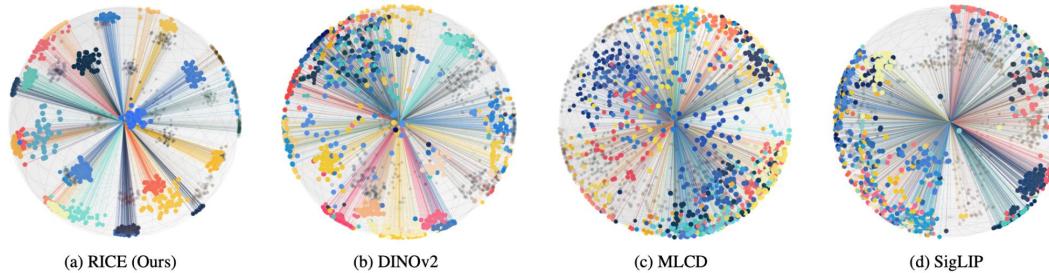
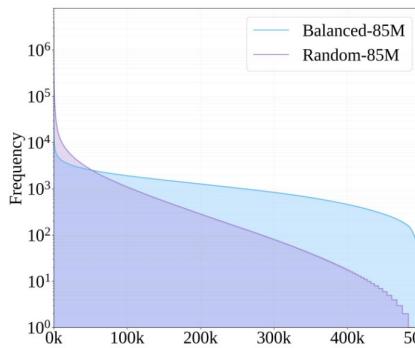


Figure 1. Visualization of object feature distributions from the COCO test dataset via t-SNE projection onto a spherical manifold.

Model Configuration		OCR & Document Understanding						General Vision Understanding									
Method	Vision Tower	InfoQA	DocvQA	ChartQA	TextvQA	OCRBench	OCRBenchV2	LiveXivVQA	OCR Avg	A12D	MMBEN	MMECog	MMEPer	POPE	RealworldQA	MMStar	Other Avg
CLIP	ViT-L-14-336px	38.9	75.2	66.5	62.5	52.5	23.0	47.4	52.3	73.2	74.6	48.0	75.6	88.8	63.7	49.0	67.6
MLCD	ViT-L-14-336px	43.5	76.5	67.8	61.7	53.1	24.0	48.4	53.6	77.0	76.4	54.1	79.9	88.7	61.1	51.0	69.7
AIMv2	ViT-L-14-336px	35.4	77.2	72.7	65.9	57.2	23.9	47.3	54.2	75.4	78.6	48.3	75.0	88.4	62.2	50.2	68.3
RICE-ViT	ViT-L-14-336px	45.2	79.2	72.3	65.9	57.5	24.1	48.9	56.2	77.9	76.6	54.6	80.7	88.5	63.1	51.8	70.5
DFN5B	ViT-H-14-378px	38.6	70.9	64.4	59.4	47.3	21.9	46.2	49.8	73.5	73.4	45.8	76.9	88.6	59.9	49.1	66.7
SigLIP	ViT-SO400M-14-384px	41.4	76.7	69.3	64.7	55.4	24.0	48.4	54.3	76.2	77.0	46.1	79.9	88.8	63.7	47.3	68.4
SigLIPv2	ViT-SO400M-14-384px	43.7	79.1	70.2	66.2	58.7	25.4	48.6	56.0	77.0	77.1	46.6	80.4	89.3	63.4	52.8	69.5
RICE-ViT	ViT-L-14-378px	48.1	82.6	75.1	66.2	58.8	25.8	49.5	58.0	76.5	77.6	54.1	79.0	89.1	62.9	51.2	70.1
SigLIPv2	ViT-SO400M-16-560px	50.2	86.2	77.4	70.2	62.7	26.5	52.9	60.9	77.0	76.5	53.5	79.9	89.3	68.2	53.1	71.1
RICE-ViT	ViT-L-14-560px	53.2	87.4	78.1	69.0	60.7	26.1	53.0	61.1	76.9	78.6	56.3	79.3	88.9	65.1	50.5	70.8
Qwen-ViT from Qwen2.5-VL 7B	ViT-H-14-560px	55.9	85.8	78.8	73.7	66.2	26.8	53.4	62.9	78.8	78.4	62.0	80.8	88.6	64.2	55.0	72.5
RICE-ViT from OV-1.5 3B	ViT-L-14-560px	53.7	87.1	81.9	73.8	73.3	30.4	53.6	64.8	80.3	79.6	58.6	82.2	89.0	67.3	56.6	73.4

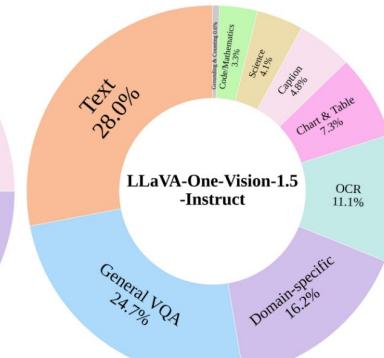
LLaVA-OV-1.5. Pre-Training Dataset



(a)



(b)



(c)

Figure 3 (a) The vocabulary coverage proportion in the LLaVA-OneVision-1.5-Mid-Traning dataset before and after concept balancing. (b) Distribution of data sources within the LLaVA-OneVision-1.5-Mid-Traning dataset. (c) Distribution of data sources within the LLaVA-OneVision-1.5-Instruct.

- Проблема: дисбаланс частот понятий в открытых датасетах
- Решение: feature-based matching, вдохновлённое MetaCLIP, но без зависимости от caption'ов
- Для каждого изображения извлекаются top-K ближайших концептов из 500К словаря MetaCLIP
- Каждому изображению назначается вес, обратный частоте его концептов

- Вероятность выбора изображения пропорциональна этому весу
- После балансировки генерируются англоязычные и китайские подписи (“powerful captioner” ???)
- Дубликаты и длинные описания удаляются
- Результат: 85M изображений с равномерным распределением концептов и языков

LLaVA-OneVision-1.5. Concept Distribution Analysis

- Анализ исходных распределений концептов показывает сильный long-tail-эффект: многие понятия встречаются редко, а небольшое число — чрезвычайно часто.
- Obelics имеет наиболее широкий и равномерный охват (наименее крутой спад хвоста).
- ImageNet-21K и SA-1B охватывают меньше концептов, но с более высокой частотой повторений.
- покрытия.

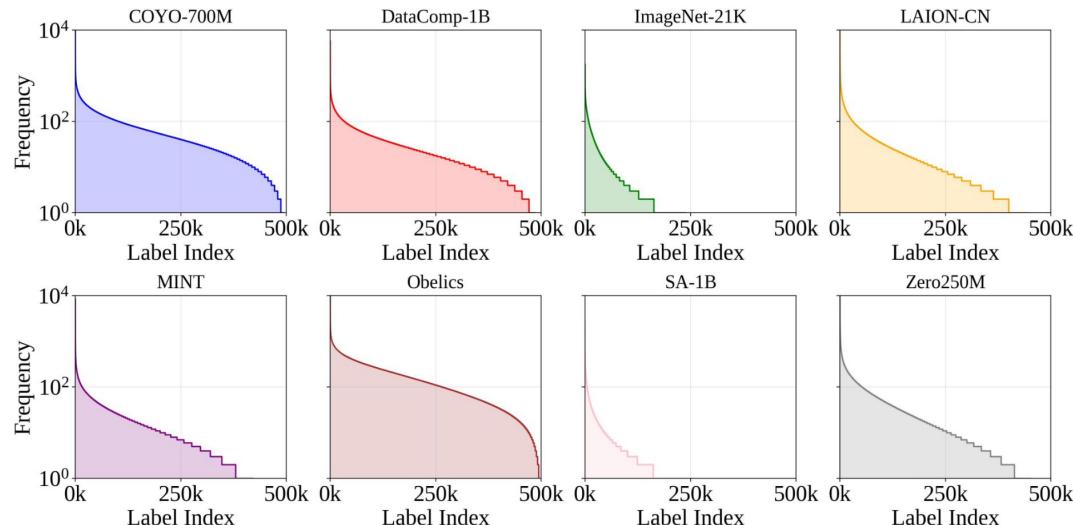
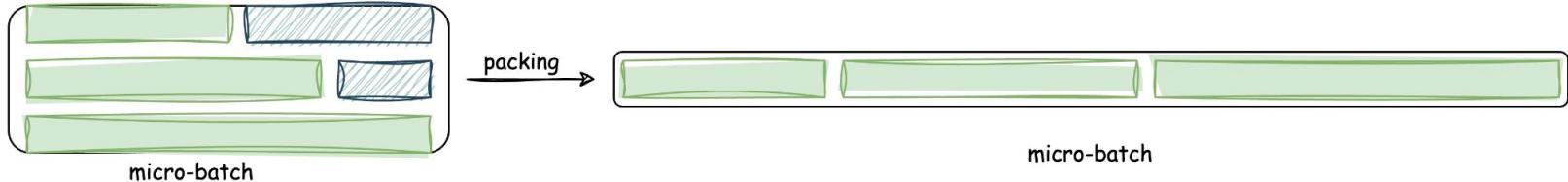


Figure 8 Original concept distributions across eight common vision datasets used in the LLaVA-OneVision-1.5-Mid-Traning dataset.

LLaVA-OneVision-1.5. Training Strategy

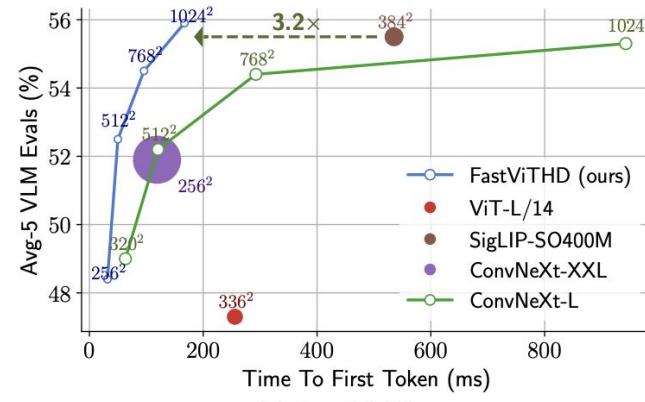


“Multiple samples are concatenated into a single long sequence to form a “packed-sample,” reducing padding (an implementation of padding-free) and improving GPU utilization.”

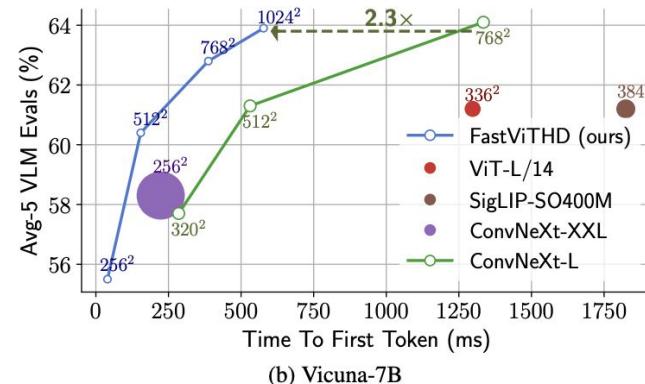
- При обучении мультимодальных моделей батчи содержат примеры сильно разной длины: короткие captions, длинные инструкции, переменное число изображений.
- Для уравнивания размеров последовательностей в batch добавляются padding-токены.
- Padding не несёт полезной информации, но занимает память и вычисления.
- В больших моделях padding может составлять 30–60% всех вычислений.
- Результат — низкая загрузка GPU (особенно при контексте > 8K токенов).
- В LLaVA-OneVision-1.5 вводится **оффлайновое объединение коротких примеров** до начала обучения.
- Примеры разбиваются по длине и модальности, затем короткие примеры последовательно «пакуются» до достижения целевой длины (например, 8192 токена).
- В LLaVA-OneVision-1.5 обучение 85M примеров занимает 3.7 дня на 128 A800 (китайская версия A100).

FastVLM: Efficient Vision-Language Models

- Цель: создать компактные и быстрые Vision-Language модели, способные работать в реальном времени.
 - Ключевой акцент — оптимизация визуального энкодера и сокращение токенов без потери качества.
 - Основная структура:
Vision Encoder (FastViT) → Token Reducer → MLP Projector → LLM (Phi-2 / Qwen2 base).
 - FastViT заменяет классический ViT на гибридный CNN-Transformer backbone.
 - В экспериментах показывают, что замена ViT на ванильный FastViT не ухудшает качество, при этом ускорение получается в 4 раза путем увеличения входного разрешения
 - Все сравнения на графике получены при инференсе на MacBook Pro с M1



(a) Qwen2-0.5B



FastVLM. FastViT

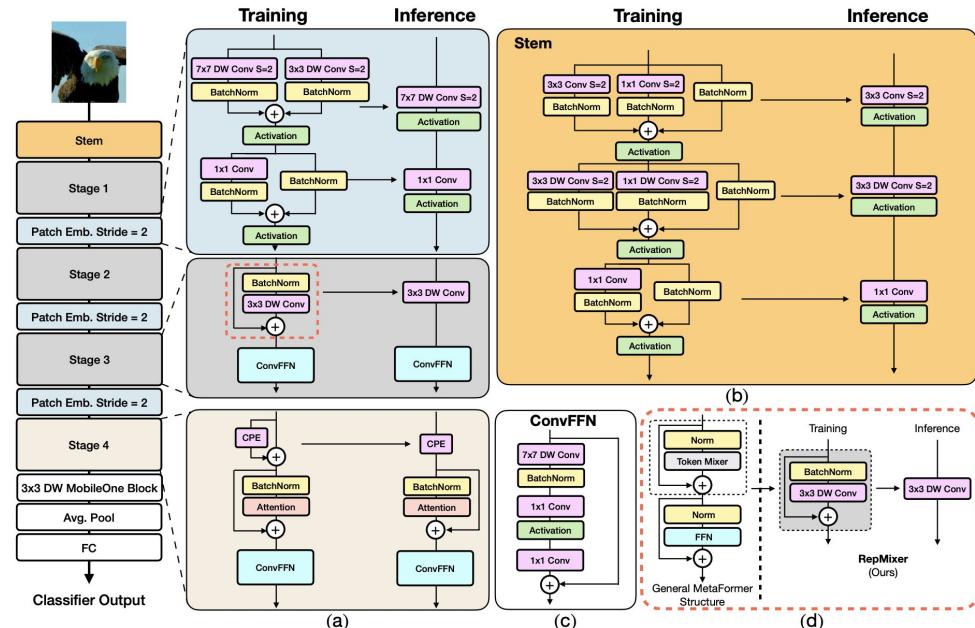
RepMixer — модуль для замены self-attention на свёрточный блок во время инференса (микс между токенами)

Во время обучения используется (BatchNorm + Depthwise Conv 3×3 + FFN). Каждый токен (патч) обменивается информацией с соседними в локальном окне (3×3) через depthwise convolution.

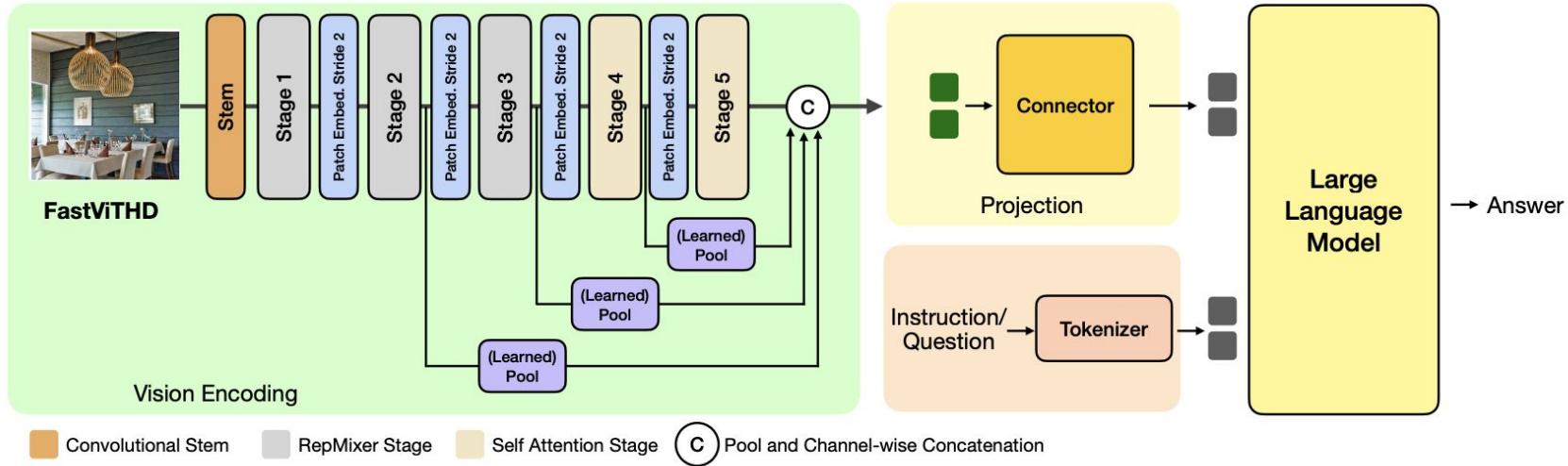
На инференсе блок “сплющивается” в одну 3×3 Depthwise Convolution с эквивалентными весами.

RepMixer сохраняет способность модели улавливать контекст, но без квадратичной зависимости от длины токенов.

ConvFFN - аналог MLP в трансформере (микс между каналами)



FastVLM. Efficient Vision-Language Models



Гибрид: ранние сверточные блоки + далее трансформер-блоки; MobileCLIP (MCi2), ~35.7M параметров.

Главное преимущество: при росте разрешения выдаёт на 5.2× меньше токенов, чем ViT-L/14

Берут multi-scale признаки через 2D Depthwise convolutions, утверждается, что работает лучше Avg.
Добавляют Stage 5 по сравнению с FastViT, чтобы еще уменьшить количество токенов

FastVLM. Static VS Dynamic Resolution

FastViTHD делался авторами так, чтобы работать эффективно на изображениях в высоком разрешении, поэтому изучили сравнение с подходами типа AnyRes.

Приходят к выводу, что в их сетапе лучше зафиксировать разрешение, чем использовать AnyRes.



Профит имеет место только на максимальном разрешении в тестировании (1536×1536).

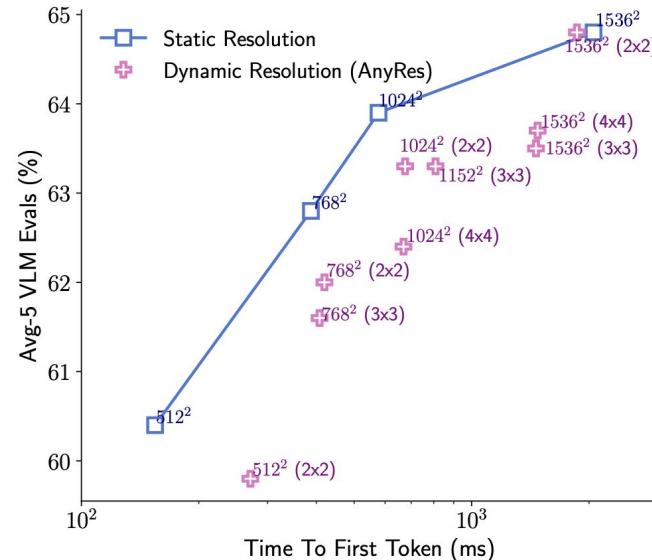


Figure 6. **Dynamic input resolution (AnyRes) is only optimal at the highest resolution when using fewer tiles (2 \times 2).** The vision encoder is FastViTHD. The tile grid size is specified in parenthesis. Training setup is LLaVA-1.5 with Vicuna 7B. Note that the *x*-axis is in log scale.

Qwen3-VL

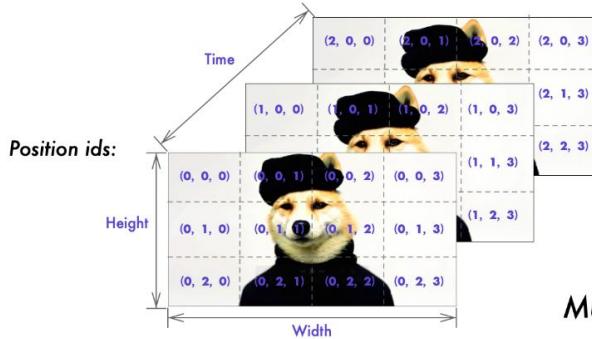


- **Visual Agent:** управляет ПК и мобильными интерфейсами — распознаёт GUI-элементы, понимает функции кнопок, вызывает инструменты и завершает задачи; топ-результаты на бенчмарках наподобие **OSWorld**.
- **Сильная текстовая составляющая:** раннее совместное предобучение текста и визуальных данных; качество на «чисто текстовых» задачах сравнимо с флагманом семейства **Qwen3-235B-A22B**
- **Visual Coding:** генерирует код по картинкам/видео — драфты → Draw.io/HTML/CSS/JS; «what-you-see-is-what-you-get»
- **Пространственное понимание:** улучшенное 2D-grounding (от абсолютных к относительным координатам), оценка позиций объектов, смены ракурса и окклюзий; поддержка 3D-grounding
- **Длинный контекст и видео:** нативно **256K** токенов, расширяемо до **1M** — можно подавать сотни страниц тэхдоков или часы видео.
- **Лучше восприятие:** расширенное покрытие объектов (себярти, аниме-герои, товары, достопримечательности, флора/фауна) благодаря более качественным и разнообразным данным предобучения.
- **OCR 2.0:** поддержка **32 языков**, устойчивость к сложным условиям (плохое освещение, блюр, наклон), рост точности по редким символам/древним письмам/терминам.



Tex. репорта пока нет

Qwen3(2.5)-VL. MRoPE



This video features a dog, specifically a Shiba

Multimodal Rotary Position Embedding (M-RoPE)

RoPE (Rotary Positional Embedding)

Вместо прибавления позиционного вектора к токену, RoPE «поворачивает» пары координат (2-мерные блоки) векторов q, k на угол, пропорциональный позиции. Это вшивает относительную позицию прямо в скалярное произведение внимания.

$$\tilde{\mathbf{q}}_i = \mathbf{R}(\omega_i p) \mathbf{q}_i, \quad \tilde{\mathbf{k}}_i = \mathbf{R}(\omega_i p) \mathbf{k}_i, \quad \mathbf{R}(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

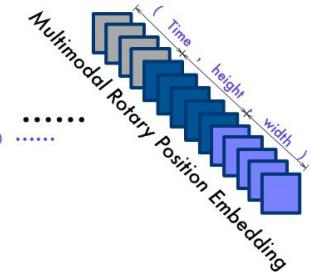
Итоговый attention-скор для токенов на позициях p и p' зависит от разности $p - p'$ (относительная позиция).

Проблема. В VLM у визуальных токенов есть пространственно-временная структура: ширина W , высота H , время T . Нужна парадигма, совместимая с 1D-текстом, но учитывающая 2D/3D-геометрию.

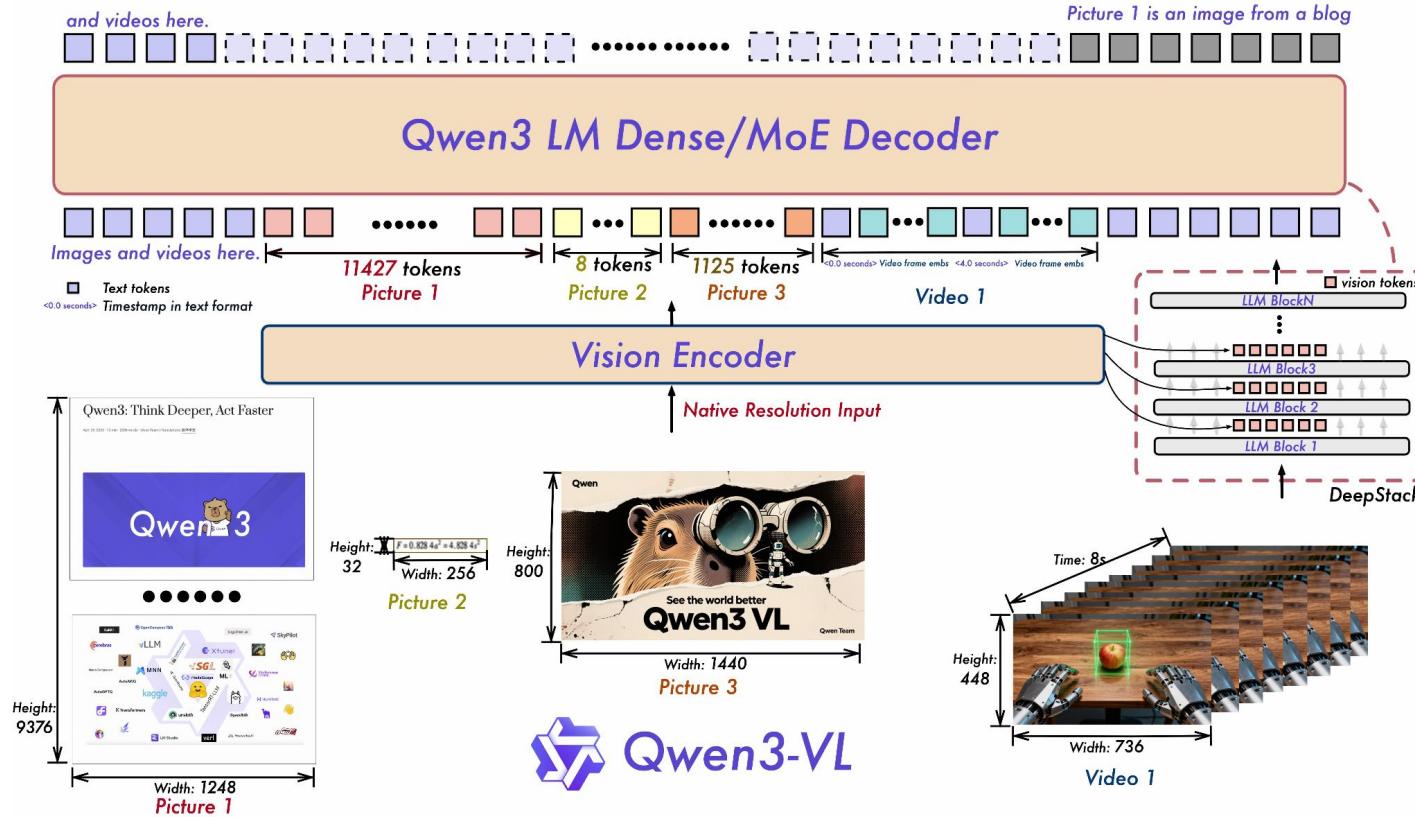
MRoPE. Разбиваем каналы эмбеддинга на «чанки» под оси t, h, w . Для пары координат, относящейся к оси $a \in \{t, h, w\}$, поворачиваем на угол $\omega_a g$, где g координаты токена:

$$\theta_i = \omega_i \cdot r_{a(i)}, \quad r = (t, h, w), \quad \tilde{\mathbf{q}}_i = \mathbf{R}(\theta_i) \mathbf{q}_i, \quad \tilde{\mathbf{k}}_i = \mathbf{R}(\theta_i) \mathbf{k}_i.$$

Для текста просто используем один и тот же ID на всех компонентах (эквивалент 1D-RoPE). Для изображений фиксирован (например, $t=0$), а h, w — по координатам патча. Для видео t растёт по кадрам, h, w — по патчу внутри кадра.



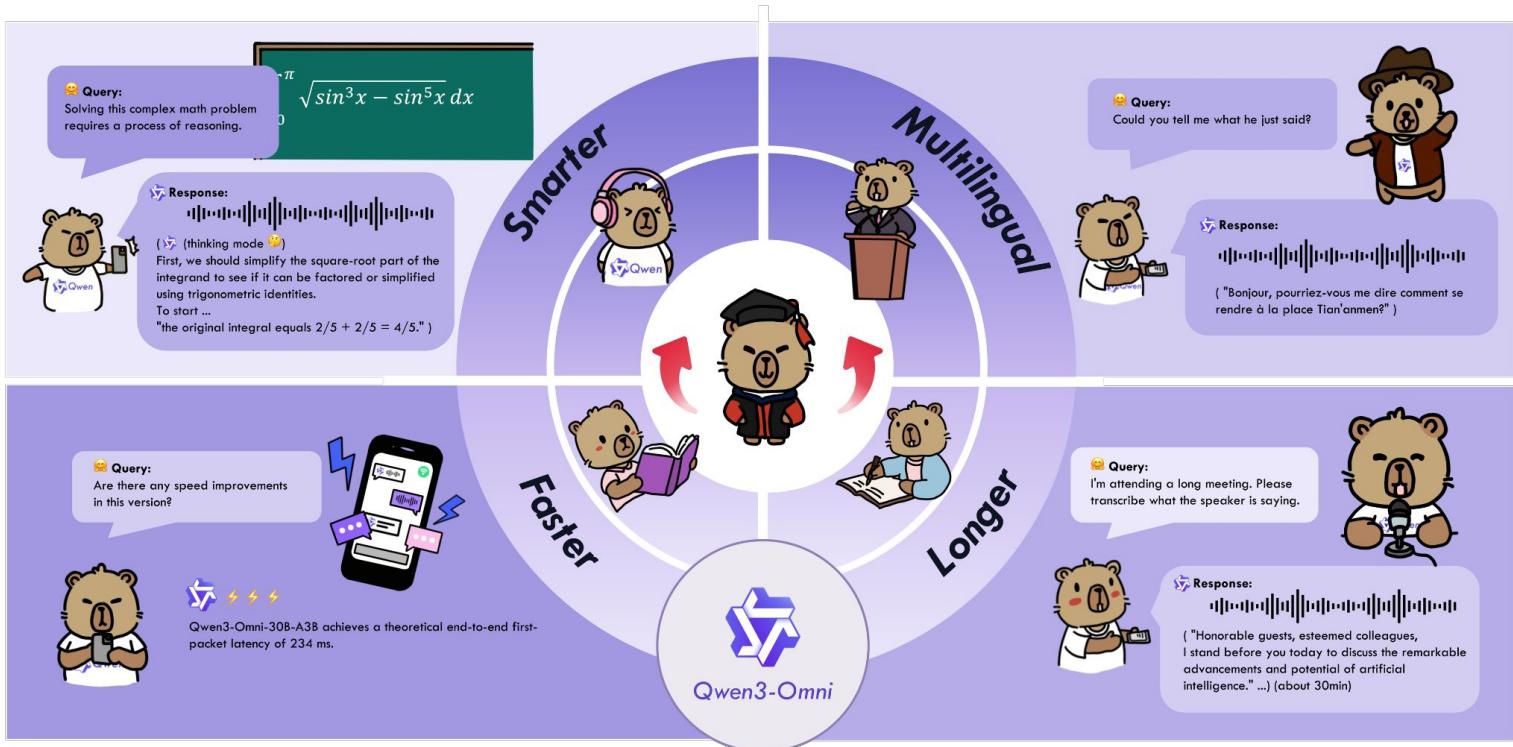
Qwen3-VL Architecture



1. Общие улучшения в VLM

2. VLM как часть модели

Qwen3-Omni



Qwen3-Omni. Общая архитектура

Qwen3-Omni следует архитектуре Thinker–Talker:

- Thinker генерирует текстовые представления и выполняет рассуждение.
- Talker получает мультимодальные (аудио + визуальные) признаки и порождает поток кодек-токенов для синтеза речи.

Главные изменения по сравнению с Qwen2.5-Omni:

MoE-архитектура для Thinker и Talker.

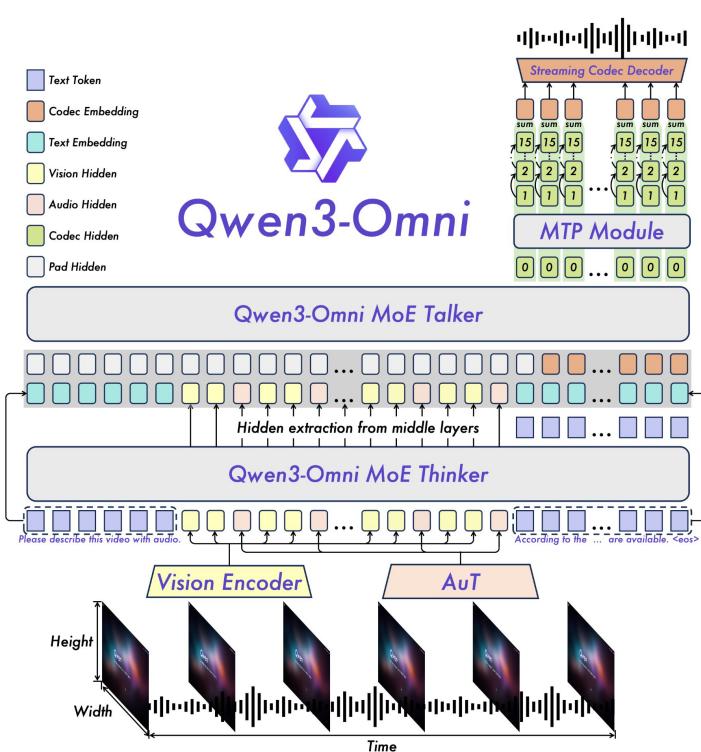
Talker больше не использует текстовые скрытые состояния Thinker, а обуславливается только на мультимодальных фичах (аудио, видео).

- Мотив: текстовые токены ≈ текстовые эмбеддинги → равноценная информация.
- Это позволяет согласовывать речь с видео / аудио (например, сохранить интонацию и тембр в переводе).

Раздельные system-prompts: стиль ответа Thinker и стиль аудио Talker можно управлять независимо.

Talker генерирует по одному codec-фрейму за шаг, MTP (“Multi-Track Prediction”) модули выдают остаточные кодбуки.

Code2Wav: лёгкая ConvNet для финального синтеза, заменяет тяжёлый DiT.



Qwen3-Omni. Audio Transformer (AuT)

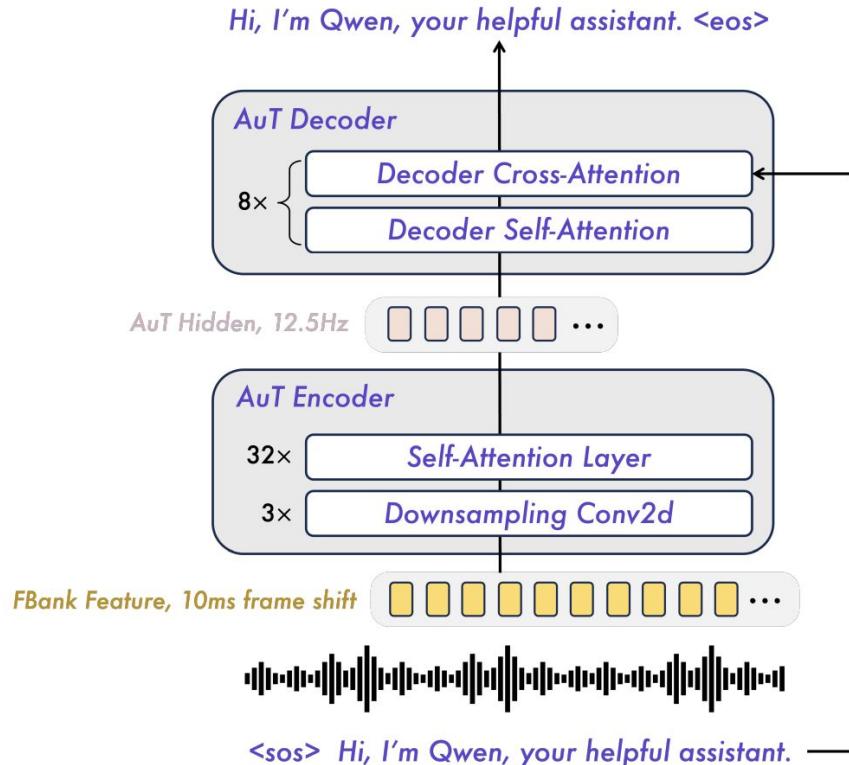
- AuT — новый аудио-энкодер на архитектуре attention-based encoder-decoder, обученный с нуля на 20 млн часов размеченного аудио.
- Используется вместо Whisper.

Архитектура и обучение

- Входные аудиопризнаки (спектральные признаки) проходят 8× даунсэмплинг через Conv2D-блоки до внимания → сокращение длины последовательности и вычислительной нагрузки.
- Далее следуют слои self-attention / cross-attention, формирующие контекстно-зависимые аудио-токены.
- Используется flash-attention с динамическими окнами внимания 1–8 секунд, при 12.5 Hz 1 s ≈ 12–13 токенов, 8 s ≈ 100 токенов
- Размер модели ≈ 0.6 B параметров.

Данные для обучения

- 20 млн часов аудио ≈ 80 % китайско-английского псевдо-ASR, 10 % ASR других языков, 10 % данные для аудио-понимания (sound event, speaker semantics и т.д.).
- Совмещённые задачи обучения:
 - распознавание речи (ASR),
 - понимание аудио (audio understanding).
- Утверждают, что такая мультимодальная постановка позволяет выучить универсальные акустические эмбеддинги, применимые к речи, музыке, шумам и аудиовидео.



Qwen3-Omni. Как преобразуются входы

Текст

- Токенизатор Qwen (byte-level BPE), 151 643 токена.
- Преимущество: устойчивость к любым символам/языкам, единое “алфавитное” пространство.

Аудио (и аудио из видео)

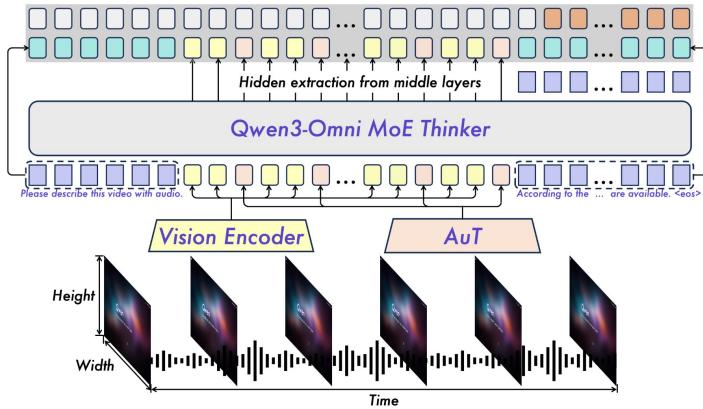
- 16 kHz, преобразование в 128-канальный mel-спектр (окно 25 мс, шаг 10 мс).
- AuT-энкодер (обучен с нуля на 20M часов): до self-attention выполняет 8x даунсэмплинг по времени.
- Итоговый token rate ≈ 12.5 Гц \rightarrow 1 аудио-токен ≈ 80 мс исходного сигнала.

Изображение и видео (без аудио)

- Vision-энкодер из Qwen3-VL, инициализация SigLIP2-So400m (~ 543 М).
- Дообучен на смеси image+video.
- Динамическая частота кадров: выбор FPS видео синхронизируется с аудио-семплингом

Единое мультимодальное представление для Thinker

- Текст \rightarrow эмбеддинги токенов.
- Аудио \rightarrow последовательность AuT-токенов (12.5 Гц, ~ 80 мс/токен).
- Картинка/видео \rightarrow визуальные токены из SigLIP2-So400m.
- Все потоки подаются в Thinker как единая интерплированная последовательность с явными маркерами модальностей



Qwen3-Omni. Обучение

Pre-train

Stage 1: Encoder Alignment

Стандартно, цель - согласовать визуальные и аудио-энкодеры с LLM (Qwen3).

– Инициализация:

- LLM — Qwen3 (Yang et al., 2025a),
- Vision Encoder — Qwen3-VL (на базе SigLIP2-So400m),
- Audio Encoder — AuT.

– Обучение:

- LLM заморожен, обучаются адаптеры и энкодеры (поочерёдно).
- Отказ от совместного тренинга encoder+adapter (как в Qwen2.5-Omni) во избежание того, чтобы энкодер “компенсировал” слабости замороженного LLM.
- Утверждают, что это улучшает “чистоту” восприятия

Stage 2: General Multimodal Training

- Полное размораживание всех параметров.
- Объём ≈ 2 трлн токенов:

Text 0.57 T, Audio 0.77 T, Image 0.82 T, Video 0.05 T, Video-Audio 0.05 T.

- Богатые межмодальные соответствия → улучшение понимания и взаимодействия между звуком, визуальными и текстовыми сигналами.

Stage 3: Long Context Training

- Увеличена максимальная длина последовательности 8 192 → 32 768 токенов.
- Добавлена большая доля длинных аудио и видео в датасете.
- Эффект: существенное повышение способности к пониманию и генерации длинных контекстов (multi-turn диалоги, видео до 40 мин).

Post-train

Stage 1: SFT (Supervised Fine-Tuning)

– Цель — перевести модель из «предобученного режима» в «инструкционный».

– Архитектура не меняется, используется малый набор данных, просто теперь instruct данные.

Stage 2: Strong-to-Weak Distillation (как в Qwen3)

Off-policy distillation — используются ответы сильных моделей-учителей (Qwen3-32B / 235B-A22B); студент учится воспроизводить их.

On-policy distillation — Thinker сам генерирует ответы на выборках промптов, выравнивает свои логиты с учителем по KL-дивергенции.

Stage 3: GSPO (Generalized Self-Play Optimization)

RL-подобное улучшение с двумя типами вознаграждения:

- Rule-based — для задач с проверяемым результатом (математика, код, точные инструкции).
- Model-based — LLM-судья (Qwen3 или Qwen2.5-VL) оценивает корректность и адекватность ответов.

Captioner

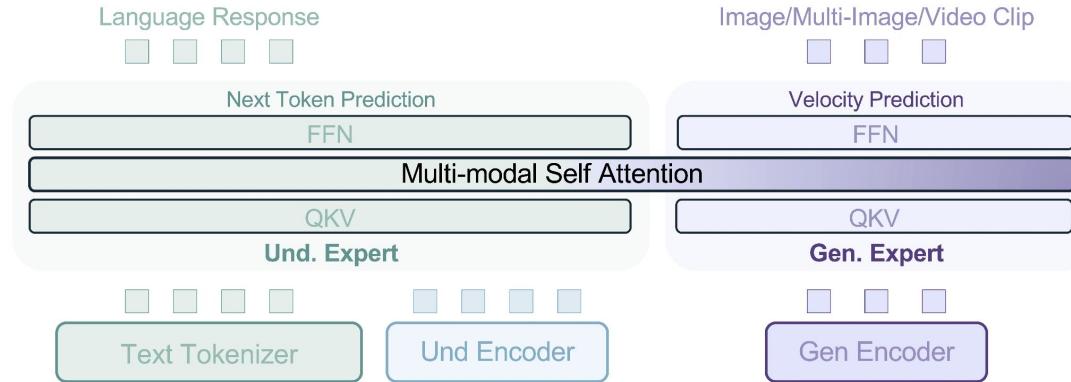
Проблема: традиционные caption-модели фокусируются на изображениях → аудиомодальность игнорируется.

Qwen3-Omni-30B-A3B-Captioner — дообученная версия

Qwen3-Omni-30B-A3B на большом корпусе аудио-описаний.

– Датасет — подробные текстовые описания акустических сцен (звуки природы, музыка, городской шум, речь).

BAGEL. Emerging Properties

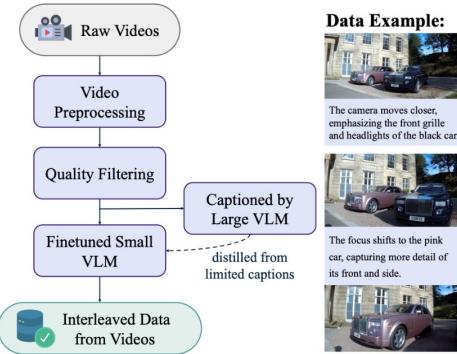


- BAGEL — универсальная unified multimodal generative model (Any-to-Any).
- Архитектура: decoder-only Transformer с двумя экспертами в MoT (Mixture-of-Transformers) — эксперт понимания и эксперт генерации.
- Нет отдельных энкодеров: все входы (текст, изображение, видео) превращаются в дискретные токены через VQ-tokenizers и обрабатываются единым attention-механизмом.
- Bottleneck-free design: понимание и генерация реализованы в одном потоке, без “энкодер–декодер” сцепки.
- Interleaved pretraining: токены разных модальностей в одной последовательности, например:
[text tokens] + [image tokens] + [text tokens] + [video tokens].
- “Emergent Properties” = модель самостоятельно осваивает новые умения: редактирование изображений, предсказание кадров, 3D-манипуляции — без явных loss-термов.

Mixture-of-Transformers (MoT):

- Understanding Expert — для текста и ViT-токенов.
- Generation Expert — для VAE-токенов (визуальная генерация).
- Оба работают на одной последовательности, общий self-attention.
- Формат тренировки:
 - Текст → Next-Token Prediction.
 - Изображение → Rectified Flow (диффузионный loss).
- Для понимания (Understanding path):
 - ViT (SigLIP2-So400m/14) → токены при 384-разрешении +2-слойный MLP соединяет ViT и LLM.
- Для генерации (Generation path):
 - VAE из FLUX (замороженный).
 - 2D RoPE позиционное кодирование для ViT и VAE токенов.

BAGEL. Данные



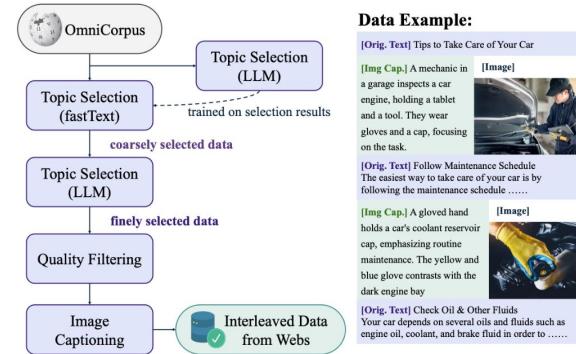
(a) Data pipeline for interleaved data from videos.

Text only data

- Поддерживает LLM компетенции: логика, рассуждение, генерация.
- Мультиязычный корпус с фильтрацией по качеству и дедупликацией.

Vision-Text Paired data

- Разделено на два потока: для понимания (VLM) и генерации (T2I).
- VLM: отфильтрованы по CLIP, разрешению и длине текста; добавлен OCR, chart, grounding.
- T2I pairs: высококачественные изображения + немного синтетики (из SD3, FLUX) в разных стилях и жанрах.



(b) Data pipeline for interleaved data from webs.

Утверждают, что пары изображение–текст недостаточны для контекстного рассуждения. Создают дополнительные данные.

Из видео:

- Видео → 4 кадра → описание изменений между кадрами (“inter-frame captioning”).
- Капшнер – Qwen2.5-VL-7B (дистиллированный), ≤ 30 токенов.
- 45 млн временных последовательностей.

Из веб-документов:

- OmniCorpus + LLM фильтрация по топикам.
- Для каждого изображения генерируется локальный caption.
- Переписывание длинных сегментов > 300 токенов через LLM summarizer.
- 20 млн структурированных interleaved веб-последовательностей

BAGEL. Reasoning-Augmented Data

[System Role Instruction]

You have the following information:

1. question image: [Place or reference the question image here]
2. question text: [Place the text of the question here]
3. answer image: [Place or reference the final answer image here]

Your task is NOT to output the final answer or the image. Instead, you must:

- Generate a “thinking” or chain-of-thought process that explains how you reason about the question.
- Provide the reasoning/analysis that leads to the answer image.
- The reasoning/analysis should include what should be changed in the answer image compared to the question image and what should be kept the same.

Below is an example of how your output should look. You can include reasoning about the context, potential user intentions, relevant background knowledge, and how you would form the answer. The length of outputs should be **around or shorter than 60 tokens**.

Example Output:

First, I notice the cat's determined action in pressing a button. To adjust for the answer, the focus shifts to expressing excitement or eagerness. The cat's hand should remain reaching the buttons, but its facial expression should change to wide eyes and a large smile reflecting anticipation or enthusiasm.

[System Role Instruction]

You have the following information:

1. question image: [Place or reference the question image here]
2. question text: [Place the text of the question here]
3. answer image: [Place or reference the final answer image here]

Your task is NOT to output the final answer or the image. Instead, you must:

- Generate a “thinking” or chain-of-thought process that explains how you reason about the question.
- Provide the reasoning/analysis that leads to the answer image.
- The reasoning/analysis should include what should be changed in the answer image compared to the question image and what should be kept the same.
- The reasoning should highlight that the input image structure and layout should be kept the same.

Below is an example of how your output should look. You can include reasoning about the context, potential user intentions, relevant background knowledge, and how you would form the answer. The length of outputs should be **around or shorter than 60 tokens**.

Example Output:

The user wants to change the background from a sunny garden to a snowy setting. The structure and layout of the pink unicorn with bubble details and sunglasses should remain unchanged. Only the environment needs modification: replacing green grass with snow and surrounding greenery with frosted, snow-covered plants while maintaining lighting coherence.

BAGEL. Обучение

4 стадии обучения:

Alignment — заморозка LLM и ViT, тренируется MLP-коннектор на captioning (изображения 378×378).

Pre-training (PT) — всё (кроме VAE) обучаемое, 2.5 Т токенов (текст + image-text + web + video), native-resolution режим.

Continued Training (CT) — повышение разрешения, увеличение доли interleaved данных (2.6 Т токенов).

Supervised Fine-Tuning (SFT) — 72.7 В токенов, высококачественные подмножества для понимания (LLaVA-OV, Mammoth-VL) и генерации.

Улучшение способностей в процессе обучения:

(a) Image Understanding — MMBench, MMVet, MathVista

– Быстрый рост до ~2 Т токенов, затем стабилизация.

– Понимание улучшается равномерно с ростом данных.

(b) Image Generation (GenEval)

– Способности генерации появляются рано (~0.7 Т токенов).

– Быстро достигается 85 % от максимального качества.

(c) Classical Image Editing (GEdit)

– Возникает после этапа 2.5 Т токенов.

– Переход к качественному редактированию при росте разрешения.

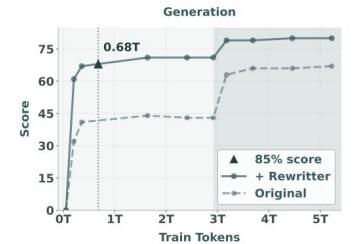
(d) Intelligent Editing (IntelligentBench)

– Самая поздняя способность (~3.6 Т токенов).

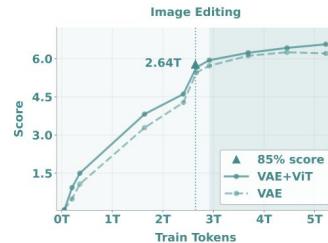
– Требует долгого контекста и мультимодальных связей.



(a) Average score on Image Understanding tasks.



(b) GenEval score on Image Generation task.



(c) GEdit Overall Score on classical Image Editing task.



(d) IntelligentBench Score on Intelligent Editing task.

Figure 7 Emerging curves. Pre-training performance curves of BAGEL on different tasks. The lighter region represents the low-resolution pre-training stage, while the darker region indicates the high-resolution CT stage. BAGEL demonstrates consistent performance improvements as the number of training tokens increases. The relationship between performance and training scale can be summarized as follows: (I) **BAGEL continues to improve** across various tasks with more training tokens; (II) **Different capabilities emerge at different stages**—understanding and generation abilities emerge first, basic editing follows, and intelligent editing emerges last, reflecting the increasing complexity of these tasks. (III) **Adopting both VAE and ViT features surpasses using VAE features alone in the image editing tasks**, especially in Intelligent Editing, with a noticeable gap. This supports the idea that ViT provides important semantic context to aid generation. Note: The average image understanding score is computed as the mean of the scores from MME-S, MMBench, MMMU, MMVet, MathVista and MMVP. All performance evaluations are conducted with BAGEL’s thinking mode disabled.

BAGEL. Models Quality VS Training Tokens

Questions	Input	1.5T	2.5T	3.5T	4.5T
Could you display what this knitting project looks like completed?					
What is the appearance of the location under night lighting?					
Can you share an image of this character looking surprised?					
Could you put some toppings on these cupcakes for me?					
Could you display the rear of this gown?					
Could you display the smoothie once it's blended?					
What method helps in adding this batter to donut molds?					

VLM2VEC / VLM2VEC-2

Проблема: универсальные мульти-модальные эмбеддинги отстают от текстовых → разрозненные модели под частные кейсы.

Идея VLM2Vec: «превратить» любой сильный VLM в эмбеддинг-модель через контрастивное обучение на большом мультизадачном датасете.

Instruction-following: эмбеддинг зависит от задачи (retrieval / VQA / классификация / grounding).

36 датасетов → 4 мета-задачи:

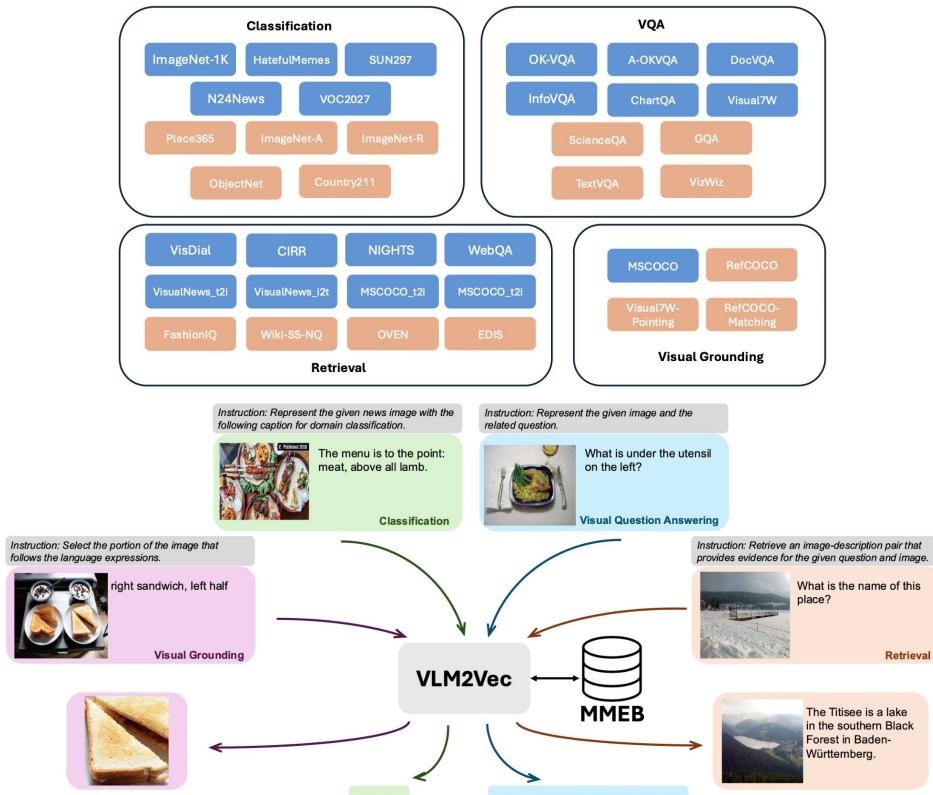
- Classification — классификация изображений.
- Visual Question Answering (VQA) — ответ на вопрос по изображению.
- Retrieval — поиск по изображению/тексту.
- Visual Grounding — локализация объектов на картинке.

Все задачи переформулированы как ранжирование, 1 истинный + негативы, среди которых происходит поиск подходящего ответа.

Единая схема входов: инструкция + запрос (текст/изображение/оба) → выбор правильного ответа.

Метрика Precision@1 — доля верных предсказаний из 1000 кандидатов.

*VLM2VEC-2 — почти то же самое, но не только для изображений.



VLM2VEC. VQA примеры из MMVB

Category	Dataset	Query Text	Query Image	Target Text	Target Image
VQA	OK-VQA (Marino et al., 2019)	<i>Represent the given image with the following question.</i> What breed of dog is this?		chihuahua	-
	A-OKVQA (Schwenk et al., 2022)	<i>Represent the given image with the following question.</i> What is the metal basket near the net used to hold?		tennis balls	-
	DocVQA (Mathew et al., 2021)	<i>Represent the given image with the following question.</i> What is name of university?		university of california	-
	InfographicsVQA (Mathew et al., 2022)	<i>Represent the given image with the following question.</i> Which social platform has heavy female audience?		pinterest	-
	ChartQA (Masry et al., 2022)	<i>Represent the given image with the following question.</i> How many food item is shown in the bar graph?		14	-
	ScienceQA (Lu et al., 2022)	<i>Represent the given image with the following question.</i> Which of these states is farthest north?		South Carolina	-
	Visual7W-telling (Zhu et al., 2016)	<i>Represent the given image with the following question.</i> Where is the man sitting?		At the computer	-

VLM2VEC. Retrieval примеры из ММЕВ

Category	Dataset	Query Text	Query Image	Target Text	Target Image
Retrieval	VisDial (Das et al., 2017)	<p><i>Represent the given dialogue about an image, which is used for image retrieval.</i></p> <p>Q:do you see a lot of people A:just 3 Q:what is the tennis player wearing A:white tennis dress Q:what color is her tennis racket A:black Q:is she wearing a hat A:a visor Q:is she close to the net A:no Q:do you see another player A:no Q:do you see a tennis bag A:no</p>	-	<p><i>Represent the given image.</i></p>	
	VisualNews_t2i (Liu et al., 2020)	<p><i>Retrieve an image of this news caption.</i></p> <p>US goalkeeper Hope Solo makes a save.</p>	-	<p><i>Represent the given image.</i></p>	
	MSCOCO_t2i (Lin et al., 2014)	<p><i>Find me an everyday image that matches the given caption.</i></p> <p>Man riding a motor bike on a dirt road on the countryside.</p>	-	<p><i>Represent the given image.</i></p>	
	WebQA (Chang et al., 2022)	<p><i>Find a Wikipedia image-passage pair that answers this question.</i></p> <p>Do both the Hays County Courthouse in San Marcos, Texas and the Ike Wood House at 227 Mitchell Street in San Marcos, Texas have six columns on their front entrance?</p>	-	<p><i>Represent the given Wikipedia image with related text information.</i></p> <p>Hays County Courthouse (2018), San Marcos, TX The Hays County Courthouse in San Marcos, Texas. Listed on the National Register of Historic Places. 227 Mitchell, San Marcos, Texas Ike Wood House at 227 Mitchell Street in San Marcos, Texas.</p>	

VLM2VEC / VLM2VEC-2. Обучение

Вход: пара (q, t⁺) — query и соответствующий target (текст, изображение или их комбинация).

К оригинальной q применяется преобразование:

$q_{inst} = [\text{IMAGE TOKEN}] \text{ Instruct: } \{\text{task definition}\} \backslash \text{n Query: } \{q\}$
чтобы модель понимала тип эмбеддинга (например retrieval или VQA).

Эмбеддинги h: берём вектор последнего токена последнего слоя модели для query и target.

Лосс L: InfoNCE контрактивный лосс с in-batch и hard негативами.
Положительная пара: (q, t⁺) должна давать высокую схожесть.
Отрицательные (N) — другие кандидаты в батче.

Функция схожести ϕ : температурно-масштабированный cosine similarity с гиперпараметром tau.

$$\min \mathcal{L} = -\log \frac{\phi(\mathbf{h}_{q_{inst}}, \mathbf{h}_{t^+})}{\phi(\mathbf{h}_{q_{inst}}, \mathbf{h}_{t^+}) + \sum_{t^- \in N} (\phi(\mathbf{h}_{q_{inst}}, \mathbf{h}_{t^-}))}$$

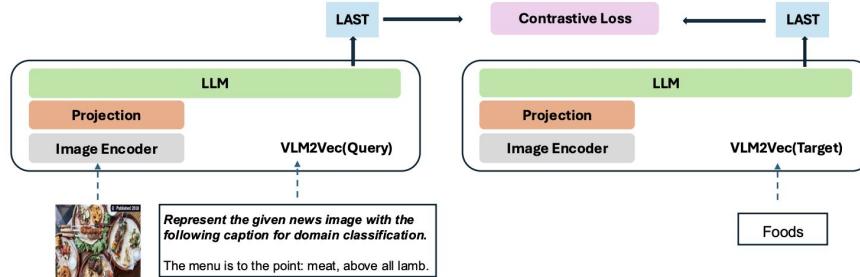
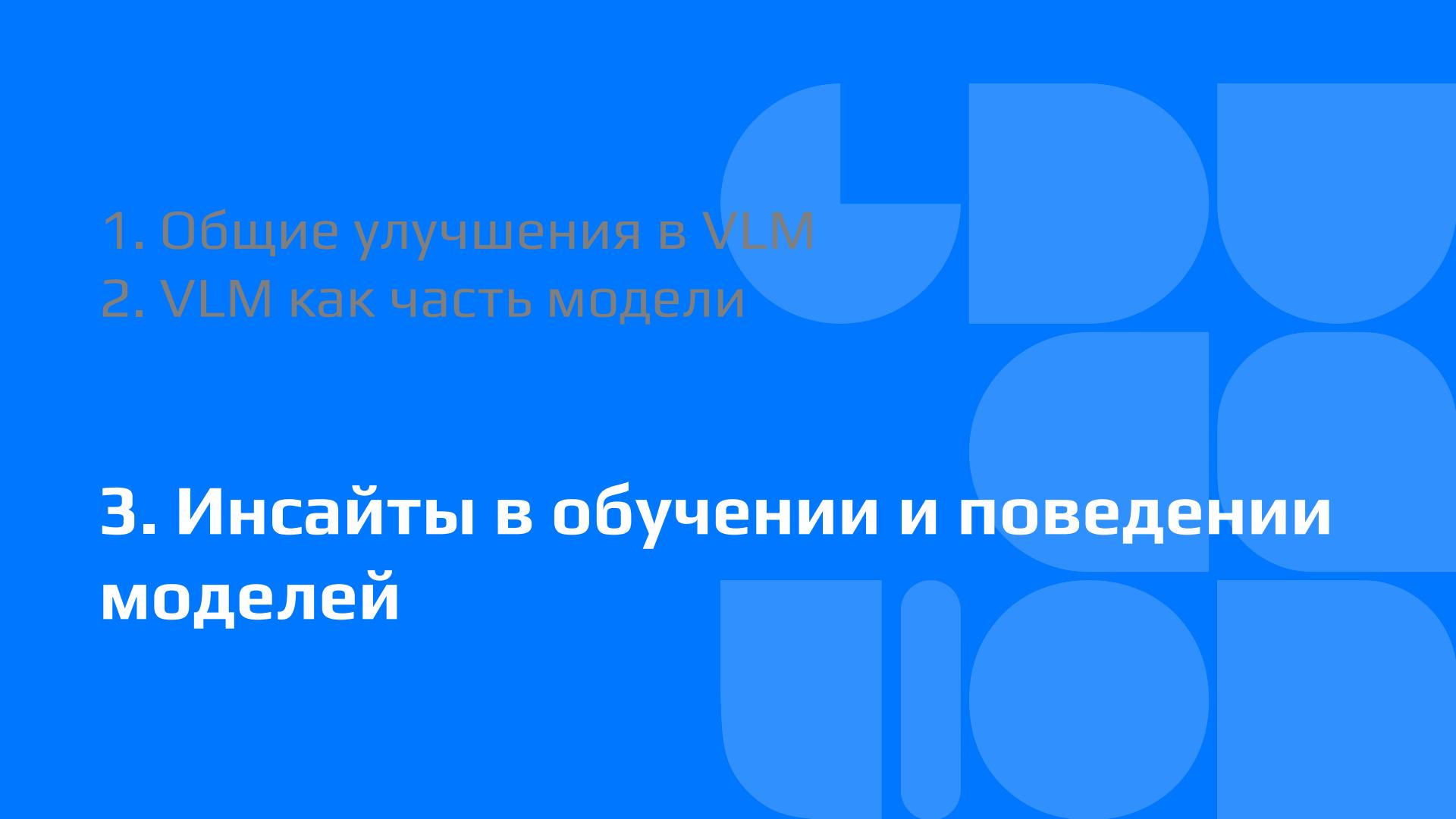
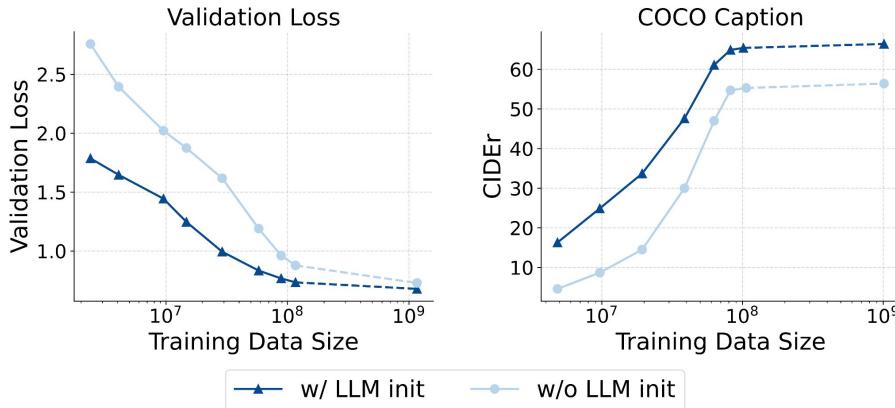


Figure 3: VLM2VEC uses a VLM as the backbone to deeply integrate image and text features. It is trained with a contrastive loss between the query and target, following task-specific instructions. The training data consists of diverse combinations of modalities on both the query and target sides, which may include images, text, or image-text pairs.

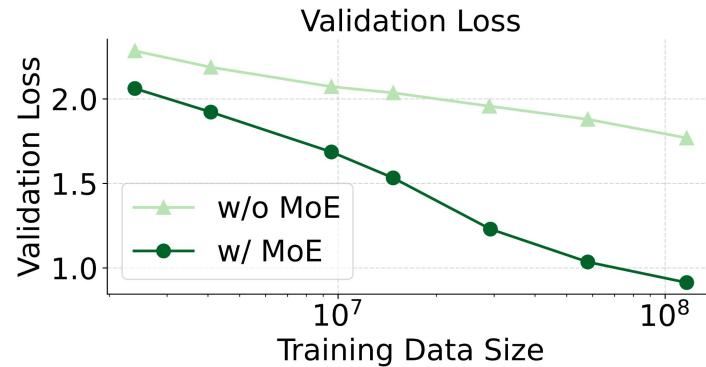
- 
1. Общие улучшения в VLM
 2. VLM как часть модели

3. Инсайты в обучении и поведении моделей

NaViL: Rethinking Scaling Properties

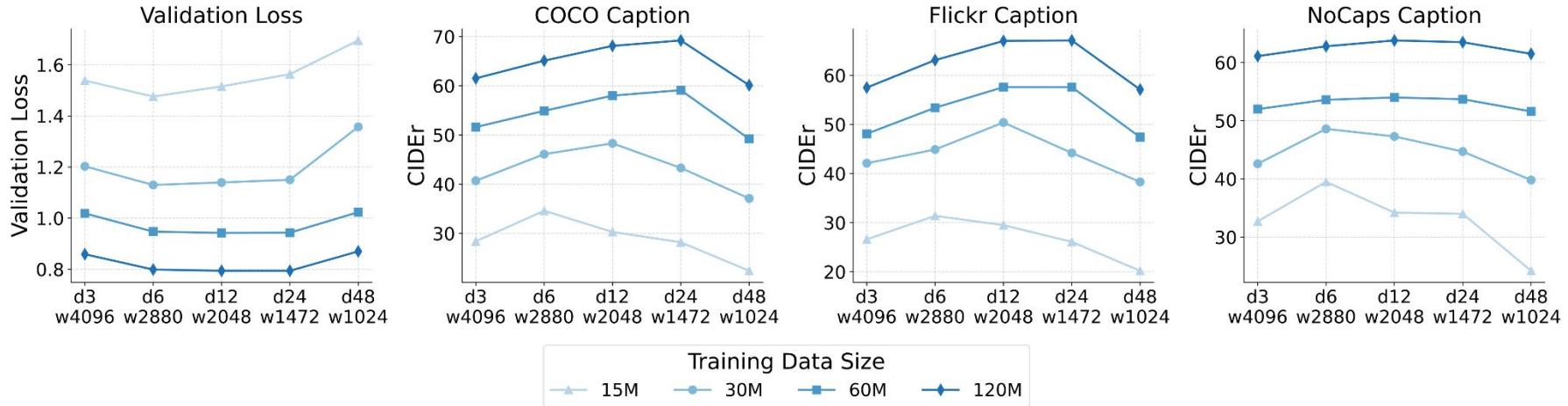


Инициализация LLM из предобученной языковой модели даёт существенное преимущество по сравнению с обучением с нуля: быстрее сходится, выше финальное качество.



Применение MoE в языковой части помогает лучше справляться с мультимодальными разнородными данными, при том что рост вычислительной нагрузки может быть не таким значительным из-за количества активных параметров.

NaViL. Flexibility of Visual Encoder Architecture

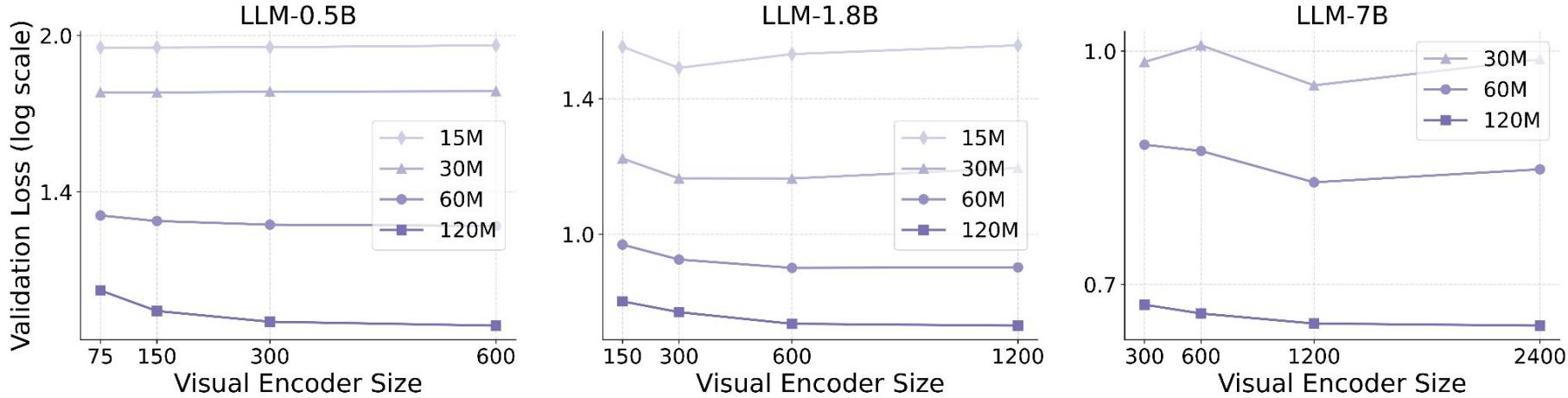


При фиксированном числе параметров визуальный энкодер показывает почти одинаковое качество при разных конфигурациях глубины и ширины.

Мелкие (более "плоские") энкодеры быстрее сходятся на ранних стадиях обучения.

Глубокие энкодеры дают небольшое, но устойчивое преимущество при большем количестве данных.

NaViL. Asymmetric Scaling Effects



Увеличение размера LLM стабильно улучшает мультимодальную производительность — по классическим законам скейлинга языковых моделей.

Рост визуального энкодера даёт всё меньший прирост качества.

Потолок качества визуального энкодера ограничивается выразительностью LLM.

NaViL. Joint Scaling Law for Vision and Language

Впервые показано (по мнению авторов): оптимальный размер визуального энкодера логарифмически пропорционален размеру LLM.

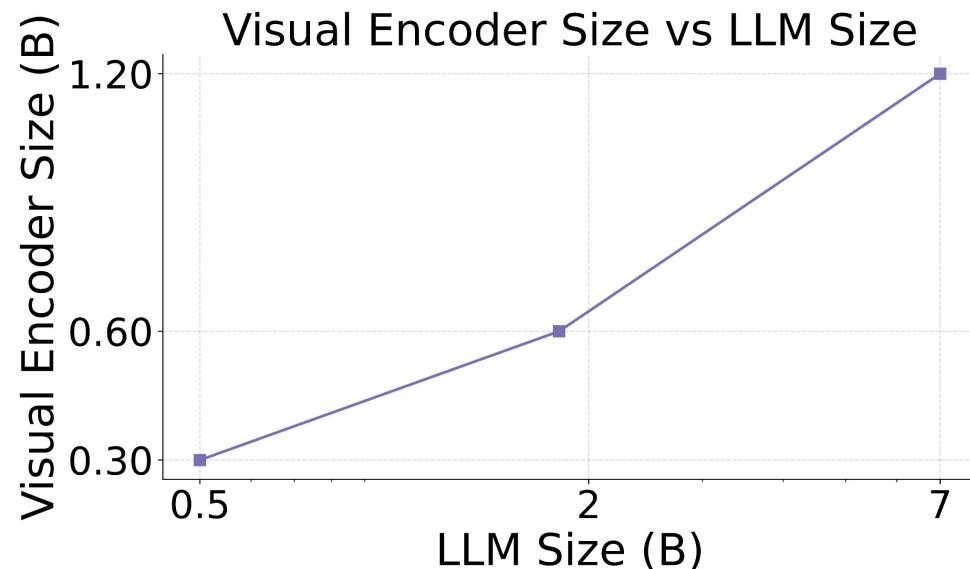
Компоненты следует масштабировать совместно, а не независимо.

Это подчёркивает неоптимальность композиционных MLLM, где фиксированный визуальный энкодер комбинируется с LLM разных размеров.

Авторы отмечают, что их исследование ограничено размерами — они не исследовали большие модели (например, >30 B параметров) и большие масштабы данных.

Фокус был на визуально-текстовых данных; другие модальности (аудио, видео, сенсоры) не покрыты.

Хочется напомнить, что это делали авторы InternVL, которые сначала сделали энкодер на 6B, а потом решились поисследовать данный вопрос.



Should VLMs Be Pre-trained with Image Data?

Работа исследует не архитектуру (она стандартная — ViT → MLP → LLM decoder, как в LLaVA), а оптимальную стратегию предобучения.

Обычно LLM обучаются полностью (3–4 Т токенов), затем начинают мультимодальный претрейн с новым LR с warmup.

Авторы показали, что это неоптимально:

- LLM уже “остывает” — глобальный LR низкий;
- мультимодальный претрейн фактически теряет эффективность.

График 3 иллюстрирует различные графики изменения learning rate, применённые в экспериментах.

- Main schedule, использовавшаяся во время исходного текстового pre-training.
- Цветные кривые — это отдельные cosine-annealing расписания, использованные на этапе обучения с изображениями и текстом.

Итоговое решение — прерывать LLM-тренировку на ~80 % пути и переходить к мультимодальному обучению без сброса LR. Так улучшаются результаты и на text-only, и на multimodal задачах.

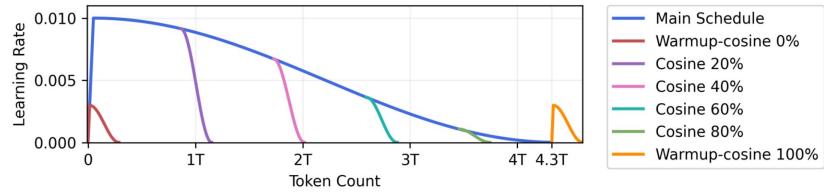


Figure 3: Representation of the different learning rate schedules used for our experiments. ‘Main schedule’ corresponds to the learning rate for the initial, text-only pretraining. Other colored schedules are the ones used for image-text training and extend over 28B tokens each. They have been upscaled and appear as extending over 280B tokens for readability.

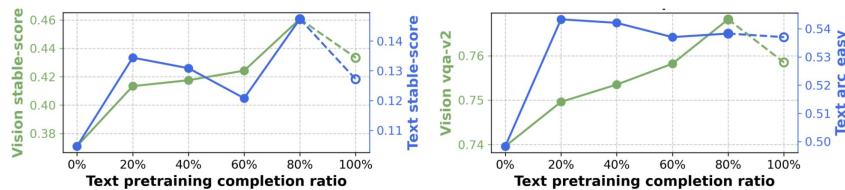


Figure 4: **Varying the length of text-only pre-training.** We analyze the impact of adding image data after varying amounts of text-only pre-training, showing results on vision benchmarks (green) and text benchmarks (blue). On the left, we show results across a suite of vision and text benchmarks; on the right, we plot two common benchmarks, VQA-v2 and ARC-easy. Introducing images at around 80% of the way through training performs best, maintaining high vision and text task performance. Note: The points at 100% are marked with hollow circles to highlight that they are trained with a different learning rate schedule, as shown in Figure 3

Импакт от визуальных данных на претрейне

Во многих работах во время стадии претрейна миксуют текстовые и мультимодальные данные, не проводя подробные исследования на эту тему.

Авторы варьируют долю картинок на стадии претрейна.

Оптимум: 10–20 % мультимодальных пар в смеси.

При большем проценте модель деградирует — “грязные” визуально-текстовые пары шумят.

При меньшем — VLM-качество не раскрывается.

Обучая с нуля - эффект от добавления негативный. Делают вывод, что тут важнее сначала запретрениваться на текстах, что согласуется с описанными ранее выводами NaViL

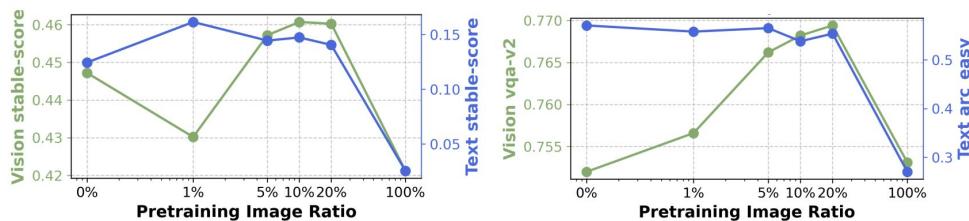


Figure 5: Varying the ratio of image to text data, after some text-only pretraining. We analyze the impact of the ratio of image to text data in pre-training, after the model has seen text-only data for most of pre-training (80%). Unlike when training from scratch (Figure 6), we find that adding vision data significantly helps vision performance, while maintaining high text accuracy.

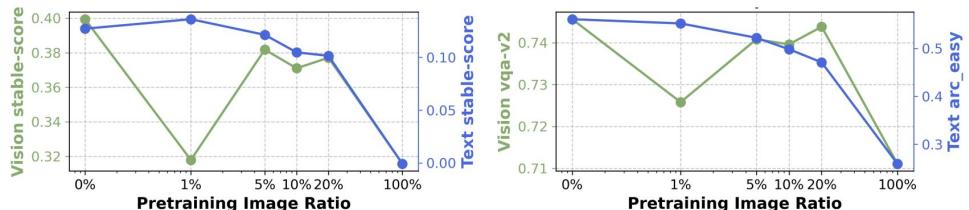


Figure 6: Varying the ratio of image to text data, when training from scratch. We analyze the impact of the image-text ratio in pre-training from scratch without any language-only pre-training. Perhaps surprisingly, when training from scratch, adding vision data consistently hurts both vision and text performance, suggesting a period of language-only training early on is important for VLMs.

Импакт visual instruct данных

Добавление инструктивных данных прямо на претрейне почти не дало эффекта (вероятно, из-за так себе LLaVA-датасетов).

На SFT-этапе оптимум ≈ 4 эпохи; больше — оверфит, опять же, на не самых чистых данных, но при высоком качестве данных больше эпох может быть полезно.

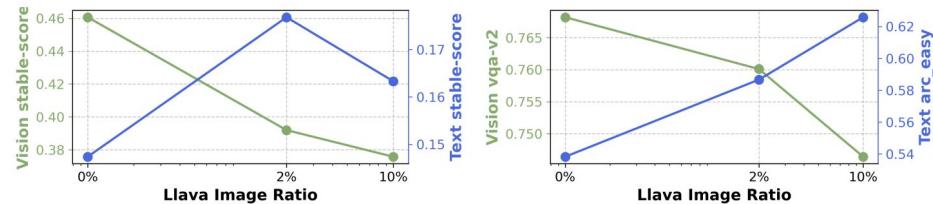


Figure 7: **Varying the proportion of instruction tuning data in the image mix.** Is including instruction tuning data during pre-training is helpful for VLMs? Surprisingly, we find that adding this data to pre-training *harms* performance. We hypothesize that this may be due to overfitting, or because mixing instruction tuning data with image-caption pairs degrades learning at this scale.

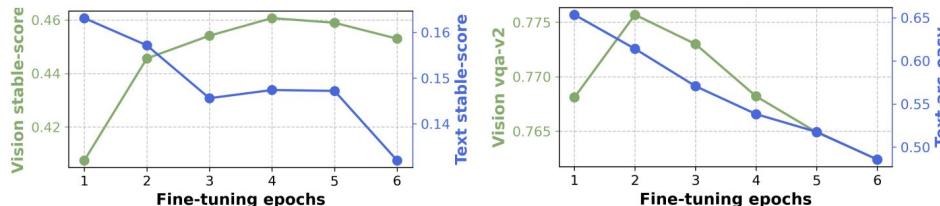


Figure 8: **Varying the number of fine-tuning epochs.** We find that fine-tuning for 2-4 epochs after pre-training performs best for vision tasks, with 2 epochs being a sweet spot for maintaining text performance while achieving high vision performance.

Image First vs Text First

Исследование Wardle & Susnjak (2024/25) показало, что порядок модальностей (изображение ↔ текст) влияет на качество мультимодального рассуждения.

Эксперименты на трёх коммерческих LLM (GPT-4o, Gemini 1.5 Flash, Claude-3 Haiku) и двух бенчмарках (M3Exam, M3COTS) с тремя схемами: Image→Text, Text→Image, Interleaved.

Для простых задач с одним изображением — схема Image→Text обеспечивала значительно лучший результат.

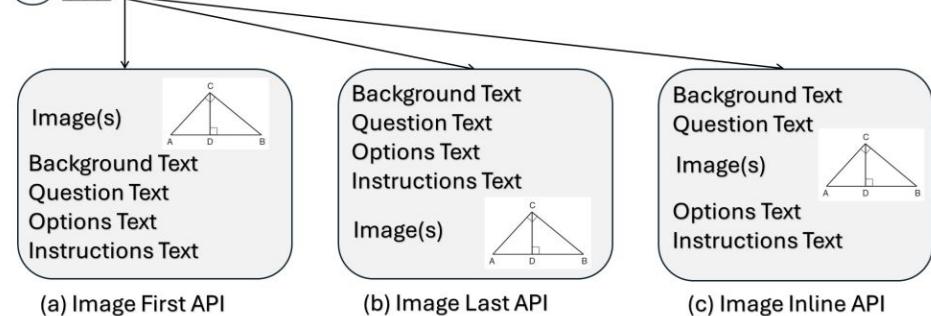
Для сложных multi-step задач эффект порядка уменьшался, критичен не только порядок, но логика подачи модальностей.

Практическое правило: структурируйте порядок модальностей так, чтобы он отражал логическую цепочку рассуждения, а не просто «картинка всегда первой».

Ограничения исследования: будем честны, мы не знаем, что происходит под капотом, поэтому следующий слайд...

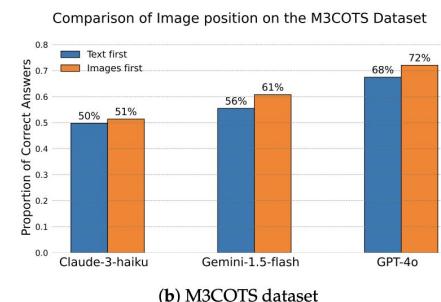
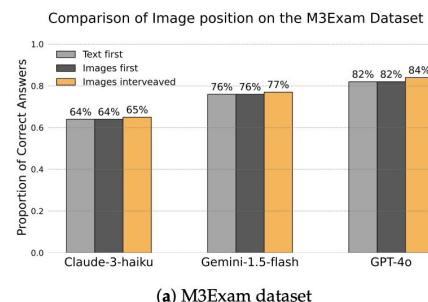
① System: “You are an expert in {subject}, helping a student answer an exam question.”

② User:



③ Instructions Text :

Think step by step to answer the question



Position Bias of Multi-image VLM

							
Question: Can you observe the differences between <image 1> and <image 2>?		Question: How many cats are there in <image 1>, <image 2> and <image 3> in total?			Question: Among <image 1>, <image 2> and <image 3>, which image looks most realistic and not synthesized?		
👉 : The first image depicts a night sky and water while the second image features a darker sky with snow covering on the ground.		👉 : There are three and two cats in the first and second image respectively, while there is only a dog in the third image, therefore five cats in total.			👉 : The first image is most natural. The second and third images depict a mural of painted robot on the building and a dumpster with fire, which displays surrealism and exaggeration.		
Question: Can you observe the differences between <image 2> and <image 1>?		Question: How many cats are there in <image 3>, <image 2> and <image 1> in total?			Question: Among <image 3>, <image 1> and <image 2>, which image looks most realistic and not synthesized?		
👉 : The two images you provided appear to be identical . There are no observable differences between them.		👉 : I can see one cat in the first image , two cats in the second, and three cats in the last one. There are totally six cats in these images.			👉 : Overall, the first image looks most realistic since dumpsters can catch fire, while the red balloon and the robot painting in the other two images render a digital and artistic style.		

Figure 1. Examples where the predictions of LVLMs are influenced by changes in the image positions (Date accessed: Oct 25, 2024). The tasks, from left to right, are difference spotting [56], counting [19] and forensic detection [15], respectively. The correct answers are highlighted in green, while the incorrect ones are highlighted in red. The prompts and outputs are simplified here for clarity and aesthetics.

В multi-image задачах предсказание **сильнo зависит от позиции** изображения во входной последовательности. Перестановка картинок меняет ответ даже у SOTA-моделей.

Position Bias of MiVLM. The Position Matters

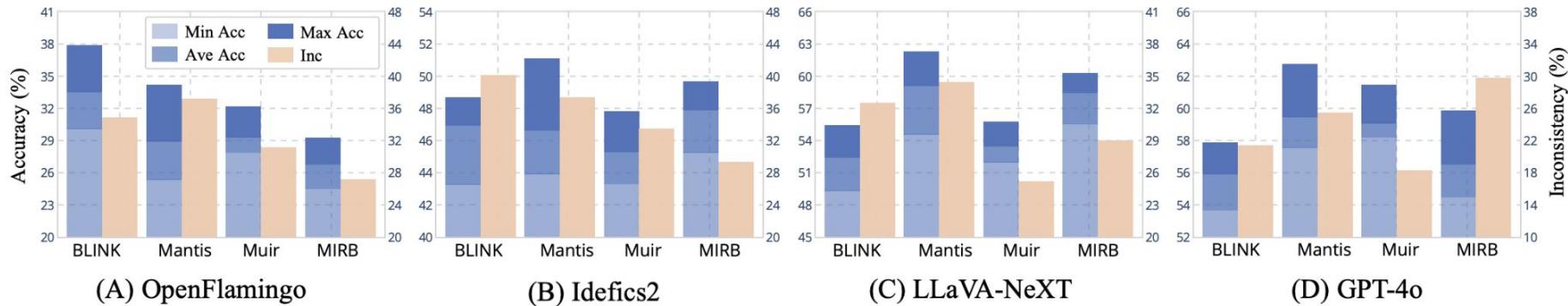


Figure 2. The results of multiple evaluations while shuffling image positions. We record the minimum, maximum, and average accuracy (left vertical axis), along with the prediction inconsistency between the best and worst-performing evaluations (right vertical axis).

Эксперимент: для BLINK, Mantis-Eval, MuirBench, MIRB многократно перетасовывают позиции изображений.

Метрики: min / max / average accuracy, inconsistency между лучшей и худшой перестановкой.

На ряде моделей расхождение >30% по accuracy, только из-за порядка картинок. GPT-4o тоже даёт до ~10%.

Position Bias. Интуиция

Что делает типичная multi-image VLM (decoder-only) под капотом

I1 — собака I2 — кот I3 — яблоко

Вопрос: «На каком изображении красное яблоко? Ответь номером.»

Каждую картинку прогоняют через vision-энкодер → получается пачка визуальных токенов: [I1-токены][I2-токены][I3-токены].

К этим токенам пришивают текстовые токены вопроса: [... визуальные токены...] + [...текст вопроса...].

Всё это — одна длинная последовательность в LLM-декодере.

В декодере causal attention: каждый следующий токен может «смотреть» только налево (на то, что уже было).

Когда модель генерирует ответ, она «смотрит» на всё, что было до этого, но её внимание естественно сильнее на последних кусках (они уже «собрали» контекст слева).

Итог: поздние картинки имеют преимущество — у них больше контекста и они «ближе» к моменту генерации ответа. Это и есть recency bias. Если нужная картинка окажется ближе к концу — шанс правильного ответа выше; если в начале — ниже.

Идея SoFA (SoFt Attention) — между визуальными токенами маску внимания делают смешанной: «чуть-чуть» добавляют двунаправленности (правые картинки становятся видимы левым), Это регулируется ручкой σ (насколько много «двунаправленности» подмешать).

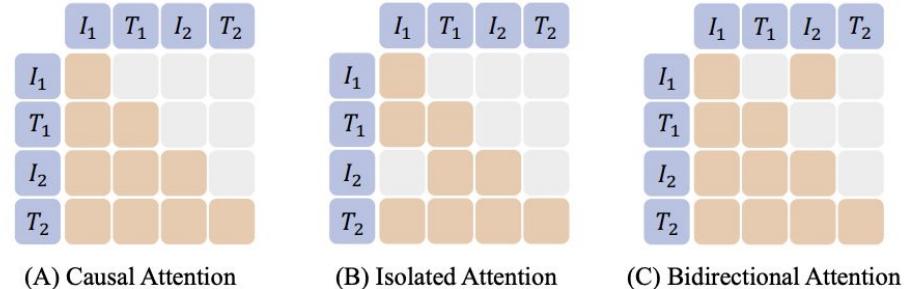


Figure 4. The three inter-image attention mechanisms, where I_1 and I_2 represent the tokens of two images, T_1 and T_2 represent the tokens of two text segments. Note that typically each image involves numerous tokens, e.g., 576 for LLaVA. Here for clarity we simplify them to a single token. In (A), images interact in a unidirectional manner, allowing I_2 to attend to I_1 , while I_1 remains isolated. In (B), each image is isolated, indicating they can only attend to themselves. In (C), bidirectional interaction is enabled so that each image can attend to any other images. It is worth mentioning that we only alter the inter-image attention while preserving causal attention between the text segments.

Look Again, Think Slowly

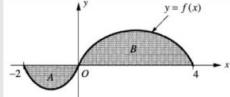
Современные мультимодальные модели получают способность к «slow-thinking» (chain-of-thought) через текст.

Однако при визуальном reasoning они зачастую теряют связь с изображением: визуальное внимание падает с ростом длины ответа.

Авторы вводят концепт visual reflection — когда модель не просто отвечает, а пересматривает визуал и проверяет ход рассуждений.

Предлагается новая модель Reflection-V, ориентированная на усиление визуальной рефлексии через два ключевых шага: (1) data construction; (2) reward design.

Question



Function $y = f(x)$ is shown in the figure. The area of region A, B are a, b respectively. If $g(x) = f(x) + 3$. what is the average value of g on the interval $[-2, 4]$?

Textual Reflection

<Think> Region A lies under the curve from $x= -2$ to 0 , while region B $x=0$ to 4 . The average value of $g(x)$ over $[a, b]$ is $\frac{1}{b-a} \int_a^b g(x) \dots$ Let's verify this calculation again... This calculation is correct. <\Think>

The final answer is $\frac{a+b+3}{6}$ X

Visual Reflection

<Think> Region A lies under the curve from $x= -2$ to 0 , while region B $x=0$ to 4 . The average value of $g(x)$ over $[a, b]$ is $\frac{1}{b-a} \int_a^b g(x) \dots$ Let's check the image again, Region A lies below the x-axis, whereas Region B lies above x-axis...<\Think>

The final answer is $\frac{-a+b+3}{6}$ ✓

Visual InformationAha MomentFinal Answer

Figure 1: Existing “slow-thinking” VLMs claimed “aha moment” is often merely a textual reflection. We instead highlight visual reflection, where VLM actively verifies and refines its reasoning based on visual inputs.

Visual Attention Weight и Dependency Measure

Для всех голов внимания вычисляют среднее внимание, направленное на визуальные токены

В начале рассуждения (первые 10–20 токенов) доля внимания к изображению $\approx 35\text{--}40\%$.

После середины цепочки падает до $< 10\%$.

То есть модель “сматривает” на картинку только в начале, а потом рассуждает чисто текстом.

Также оценивают, насколько распределение вероятностей следующего токена меняется при удалении изображения после нескольких сгенерированных токенов.

Используется расстояние Хеллингера между распределением до и после удаления картинок.

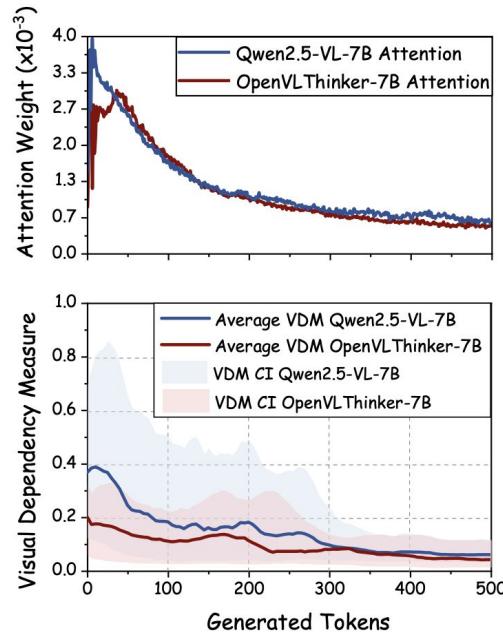
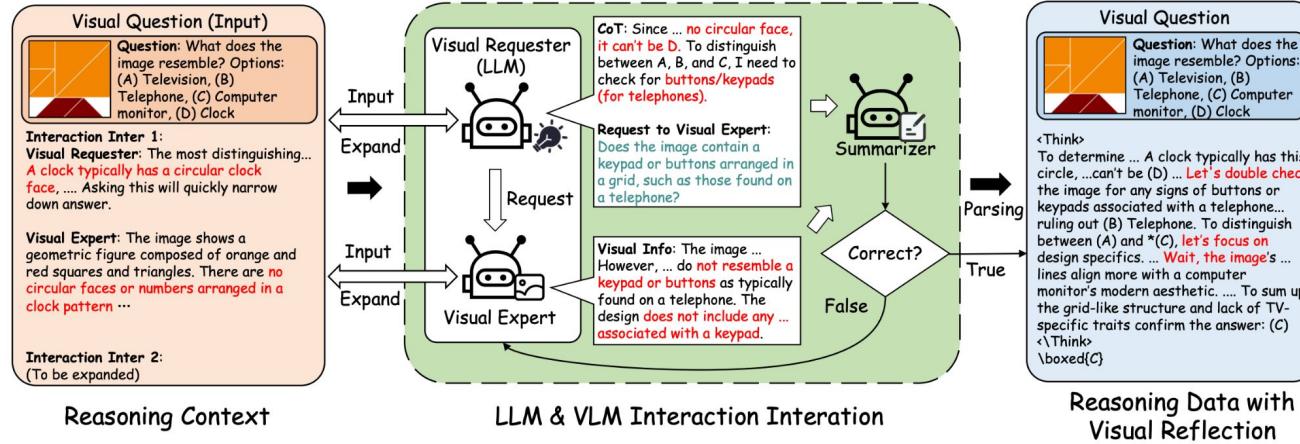


Figure 2: Attention weights on visual tokens and the visual dependency measure during reasoning on the MMMU dataset. Both metrics decline sharply as more tokens are generated, and RL-enhanced models (e.g., OpenVLThinker-7B) do not mitigate this decay.

Reasoning Data with Visual Reflection Construction



Visual Requester (LLM): формулирует, какую часть изображения нужно уточнить, и отправляет запрос VLM.

«Does the image contain a keypad or buttons arranged in a grid, such as those found on a telephone?»

Visual Expert (VLM): отвечает, описывая нужный визуальный фрагмент; эта информация добавляется в контекст рассуждения.

«.. do not resemble a keypad or buttons as typically found on a telephone. The design does not include any ...»

Summarizer (LLM): объединяет всё рассуждение и проверяет ответ. Если ответ неверный → начинается новый раунд запросов и ответов.

Такой цикл формирует примеры, где модель многократно возвращается к изображению — эффект visual reflection.

Reasoning Data with Visual Reflection Construction

Второй элемент: на этапе RL используется reward модель, ориентированная на визуальное внимание — чем больше модель смотрит на визуальные токены на протяжении рассуждения, тем выше награда.

Модель Reflection-V обучается с миксом supervised + RL, где reward включает метрику "average visual attention per reasoning step".

Это стимулирует не просто генерировать текст, а включать изображение в ход мышления.

Архитектурно: визуальные токены остаются в потоке, но за счет лосса усиливается их роль через reward signal.

Проводят ablation: и без sft, и без RL - хуже.

Input

Fig. 1

OpenVLThinker-7B

think\nThe sawtooth waveform can be represented as a Fourier series, which is a sum of sine waves. The general form of the Fourier series for a function $f(t)$ is given by: $f(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos(\frac{2\pi nt}{T}) + b_n \sin(\frac{2\pi nt}{T}))$. For a sawtooth wave, the coefficients (a_n) are zero because the wave is an odd function, and coefficients b_n can be calculated using the formula for Fourier series coefficients for an odd function: $b_n = \frac{2}{\pi} \int_0^{\pi} f(t) \sin(\frac{2\pi nt}{T}) dt$. For the sawtooth wave, the period $T = 2\pi$, and the function $f(t)$ is $f(t) = \frac{2t}{\pi}$ for $-\pi \leq t < \pi$...

think\nAnswer:D

Reflection-V-7B

think\nTo determine the correct Fourier series, we first analyze its periodicity. The image shows the waveform repeating every $T = 2\pi$. The Fourier series for a periodic function $f(t)$ with period T is given by: $\sum_{n=-\infty}^{\infty} c_n e^{i n \omega_0 t}$, where $\omega_0 = \frac{2\pi}{T}$. For a sawtooth waveform, the fundamental frequency is $f_0 = \frac{1}{T} = \frac{1}{2\pi}$, so $\omega_0 = 1$. The coefficients c_n are determined by the function integral over one period. The Fourier series starting at $-\pi$ with a slope of 1 is: $\sum_{n=1,3,5,\dots}^{\infty} \frac{(-1)^{n+1/2}}{n} \sin(nt)$.

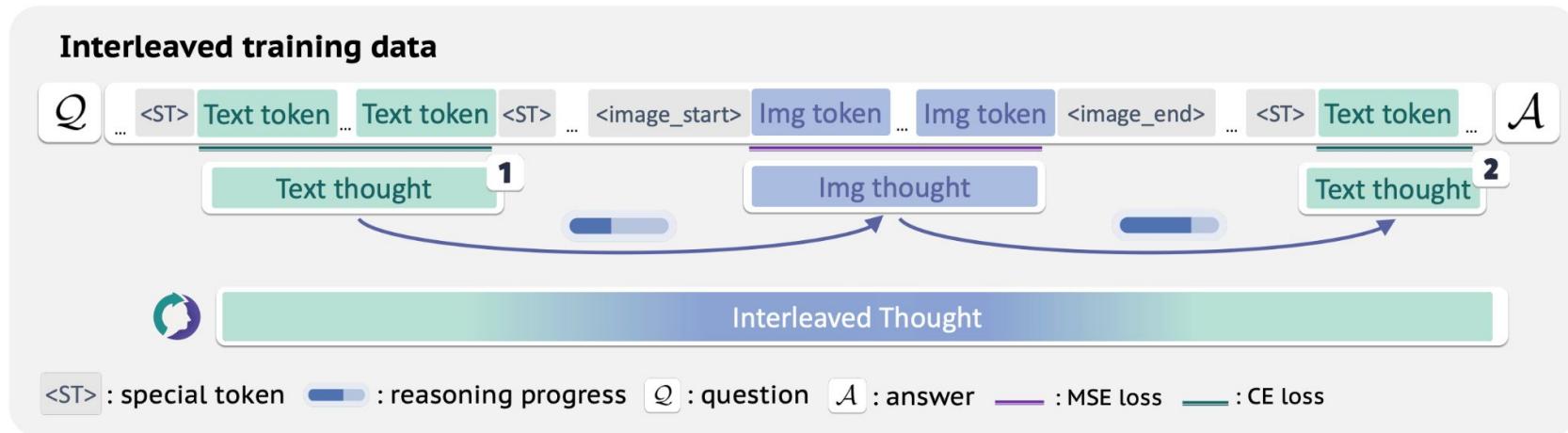
Visual Aha Moment!

*Let's check the image again.\n*The image shows a sawtooth waveform with a period of 2π , as indicated by the horizontal axis labels (0, π , 2π , 3π , ...). The waveform starts at $-\pi$ and increases linearly to 0 at π ($t = \text{lpV}$), then decreases back to $-\pi$ at $t = 2\pi$, and repeats ...**

think\nboxed{C}

Figure 5: Our model, Reflection-V, exhibits the visual reflection capacity that we claim, in contrast to other RL-based visual reasoning models. The background color of tokens in the figure indicates the magnitude of the visual attention weight. This visual reflection capacity is demonstrated by the recheck and attention to visual tokens again that appear alongside "Aha moment", like "Let's check the image again".

ThinkMorph



Текстовый СоT не покрывает случаи, где нужны **манипуляции с визуальным контентом** (поиск, навигация, перестановка частей, фокус на графике). В **ThinkMorph** предлагают относиться к тексту и изображению как к **комплементарным** каналам, а не как к взаимозаменяемым (isomorphic) представлениям.

Единая модель (на базе BAGEL-7B) с возможностью порождать **чередующиеся мысли**: текстовые токены и токены изображений.

Loss: MSE по токенам изображений + NLL по текстовым.

ThinkMorph. Multimodal Reasoning

Jigsaw Assembly — перестановка/сборка частей; визуализация перестановок.
Spatial Navigation — построение пути в лабиринтах; стрелки/маршруты поверх изображения.
Visual Search — подсветка боксами целевых регионов; строгая фильтрация сложности (объект 1–30% площади).
Chart Refocus — подсветка/оверлеи на диаграммах.

Unseen manipulations: зум, мульти-боксы, инпейнинг, кроп, трансформации перспективы, прогноз движения — появляются на OOD, до 10% всех операций.

Автономный выбор режима: модель иногда сама уходит в text-only, получая лучшие ответы на части кейсов.

Jigsaw Assembly

The image below is a 2x2 jigsaw puzzle... Your task is to determine the correct arrangement ...

A: ...
B: ...
C: ...
D: ...

<think>... analyze the visual content of each piece and determine their correct positions based on natural image flow ... </think>

<think>... There are no visible discontinuities ... harmoniously unified... </think>

<answer> C</answer>

Spatial Navigation

Your goal is to guide a player from the start to the goal on a grid map while avoiding holes ... Provide your solution as a sequence of moves ...

<think>...Let's analyze the maze step by step: Step 1: Identify the grid and obstacles. The player starts at the top-left corner (let's call this position (1,1)).</think>

<think>...carefully verify the path I plotted. The red arrows start at the top-left corner ... </think>

<answer> \boxed{D,D,R,R,R}</answer>

Visual Search

What is the color of the man's cap?

A. green B. white
C. orange D. blue

<think>....The red bounding box should be placed around the man's head and cap...</think>

<think>... now highlighting the man ... focusing on the highlighted area, ... the man's cap is gray, so the correct answer is C.</think>

<answer> C</answer>

Chart Refocus

Which attitude represent the smallest gap between women and men?

<think>To answer ... I focus on these two specific bars and labels ... </think>

<think> ... I analyze the highlighted area in the visualization that focuses on the second category: 'I don't discriminate against ...' </think>

<answer> C</answer>

Emergent Properties

1 Unseen Visual Manipulations

Is the bell pepper in the picture red or yellow?
A: Red B: Yellow
(From MMVP)

<think>To determine whether the bell is red or yellow, I need to closely examine the color of the pepper shown ...

<think>... I closely examine the image provided, which features a close-up shot as its central focus... Based on this detailed visual inspection, I can confidently conclude that the bell pepper is yellow... </think>

<answer> B</answer>

2 Autonomous Mode Switching

Interleaved Thought
OR
Text thought
e.g., 5.3% Switch 7.29% Gain

3 Superior Test-Time via Diversified Thoughts

Strong Scaling +8.0%
+2.7%
-2.0%

N	Text Reasoning	Visual Reasoning	ThinkMorph (Interleaved)
N=1	65	50	65
N=2	65	50	65
N=4	70	50	75
N=8	70	50	78

Vision Space
Text Space
Interleaved Reasoning Broader Space ✓

A Survey of State of the Art Large VLMs

- От «мультимодальных» к «omni/real-time». Крупные релизы 2025 года декларируют мультимодальность «из коробки».
- Почти везде: энкодеры модальностей + коннекторы к LLM-декодеру / MoE; растёт интерес к общему токен-пространству и согласованию между модальностями.
- По факту трехстадийное обучение как стандарт. Претрейн → Инстракт → предпочтения, как в LLM.
- Широкое применение синтетически сгенерированных данных для покрытия длинного хвоста задач с использованием существующих моделей.
- Эффективность и стоимость — главный драйвер. Комбинации MoE, дистилляции, QLoRA/адаптеров и компрессии контента.
- Потребность в устойчивом мультимодальном reasoning и планировании.

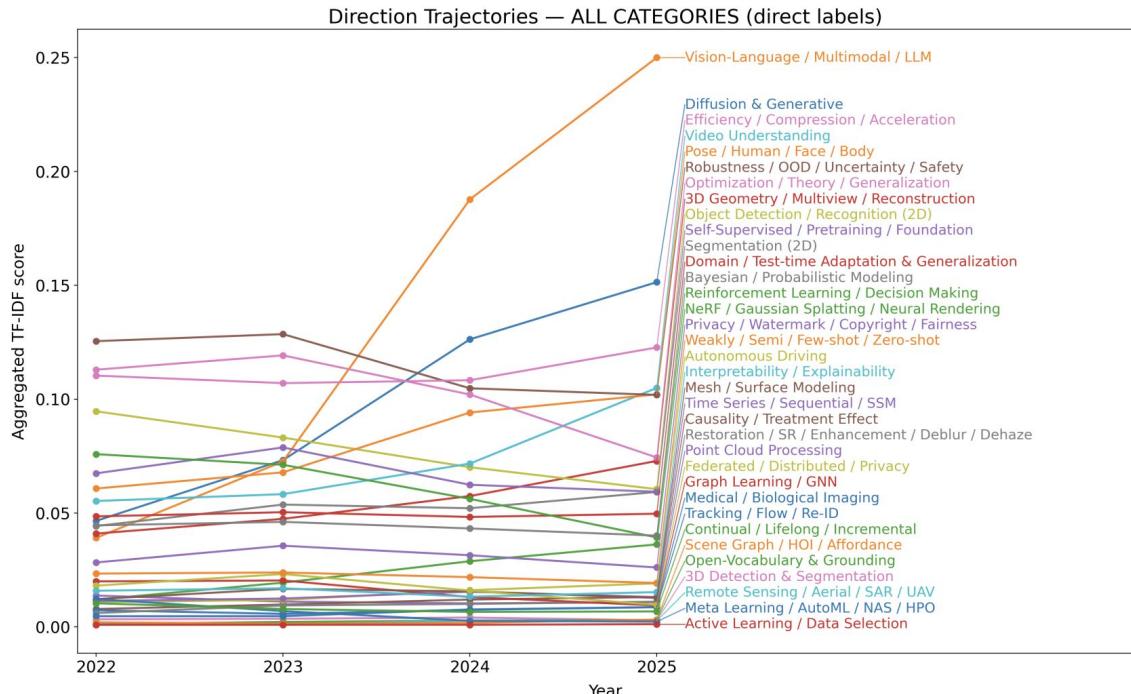


Figure 1: Direction Trajectories across CVPR+ICLR+NeurIPS — ALL CATEGORIES (direct labels). Each curve is the yearly aggregated TF-IDF mass for a direction (integer year ticks).

Спасибо
за внимание!

