Мультимодальные модели на основе Transformer

Мурат Апишев (mel-lain@yandex.ru)

Сентябрь, 2023

Transformer и не-текстовые данные

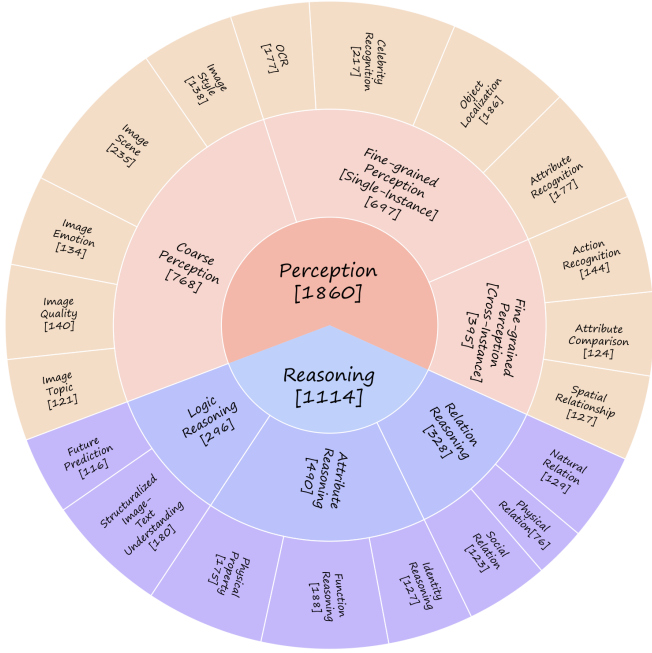
- **Этап 1:** Применение моделей на основе Transformer для разнообразных текстовых задач
  - BERT
  - GPT-2
- **Этап 2:** Адаптация успешной архитектуры к обработке других модальностей для решения дискриминативных задач и генерации признаков (CV, ASR, ...)
  - CLIP
  - HuBERT
- **Этап 3:** Генерация текста композитной моделью, принимающей на вход объекты различных модальностей
  - Flamingo
  - BLIP-2
- **Этап 4:** Добавление возможности генерировать объекты разных модальностей
  - NExT-GPT

Примеры мультимодальных задач

- Близость между текстом и изображением / аудиозаписью (в т.ч. для классификации)
- Определение связи между объектом на изображении и описывающими его словами из описания (Visual Grounding)
- Ответы на вопросы по изображениям (VQA)
- Рассуждения по изображениям и их описаниям (Visual Reasoning)
- Генерация описаний к изображениям (Image Captioning)
- Распознавание символов / фонем по аудиозаписи
- Генерация изображений по тексту [и изображению]

Пример мультимодального бенчмарка

- Есть много мультимодальных бенчмарков: ScienceQA / NLVR2 / COCO / ...
- Один из наиболее свежих и полных — MMBench:
  - английский язык
  - 20 задач
  - $\approx$ 3K примеров

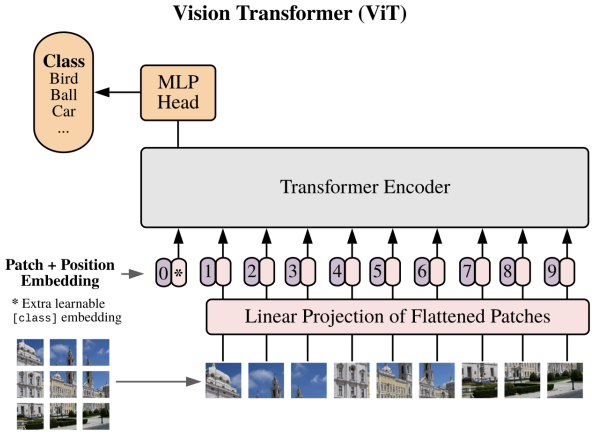


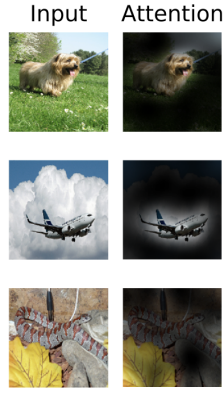
Этап 2

ViT, 2020 (Google)

- Кодировщик Transformer, на входе изображение, на выходе — его класс
- **Вход:**
  - изображение разбивается на квадраты (патчи)
  - каждый патч кодируется линейным слоем в вектор (альтернатива — CNN)
  - к вектору патча добавляется обучаемый позиционный эмбеддинг
- В CLS-токене на предобучении предсказывается класс изображения, на дообучении голова заменяется на новую
- При дообучении на целевые задачи повышается размерность изображений $\Rightarrow$ увеличивается длина последовательности
- Эмбеддинги для новых позиций получаются 2D-интерполяцией уже обученных с учётом координат патча на изображении

ViT, 2020 (Google)





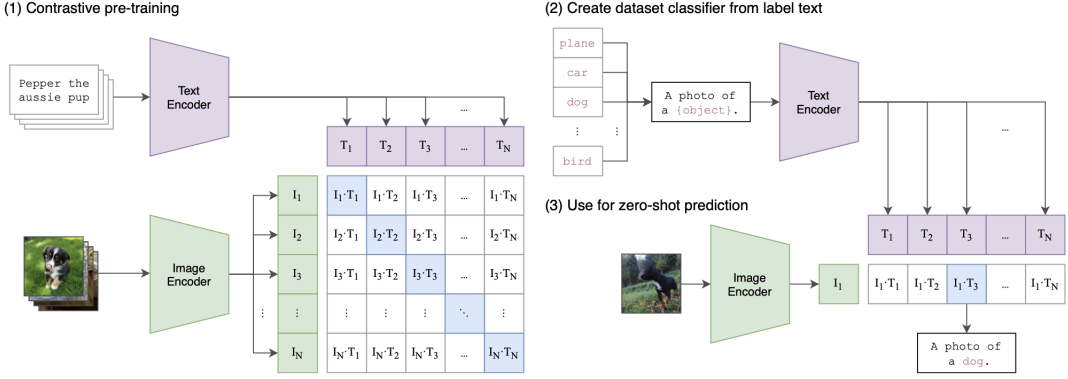
Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

CLIP, 2021 (OpenAI)

- Zero-shot классификатор изображений, регулируемый текстовыми промптами
- Два кодировщика:
  - для изображений (ResNet с вниманием или ViT)
  - для текста (GPT-2 с контекстом 76)
- На выходе каждого один вектор: для изображения и для текста его класса/описания
- Они переводятся обучаемыми линейными слоями в общее пространство и нормализуются
- На обучении между всеми парами «текст»-«изображения» считается матрица близостей и применяется CE-loss по обеим её размерностям

CLIP, 2021 (OpenAI)

- На инференсе модели подаются изображение и N текстов-промптов вида «это собака/кот/машина/. . . »
- Вектор изображения умножается на векторы всех промптов и ответ выбирается по лучшему значению близости
- Работает в zero-shot хорошо на доменах и классах из обучения



Почему эти работы до сих пор важны

На чём тестировался MMBench:

VLM	Language Backbone	Vision Backbone	Overall Parameters	Trainable Parameters
OpenFlamingo	LLaMA 7B	CLIP ViT-L/14	9B	1.3B
OpenFlamingov2	MPT 7B	CLIP ViT-L/14	9B	1.3B
MMGPT	LLaMA 7B	CLIP ViT-L/14	9B	22.5M
MiniGPT-4	Vicuna 7B	EVA-G	8B	11.9M
MiniGPT-4-13B	Vicuna 13B	EVA-G	14B	11.9M
PandaGPT	Vicuna 13B	ImageBind ViT-H/14	14B	28.8M
VisualGLM	ChatGLM 6B	EVA-CLIP	8B	0.2B
InstructBLIP	Vicuna 7B	EVA-G	8B	0.2B
InstructBLIP-13B	Vicuna 13B	EVA-G	14B	0.2B
Otter-I	LLaMA 7B	CLIP ViT-L/14	9B	1.3B
LLaVA	LLaMA 7B	CLIP ViT-L/14	7.2B	7B
LLaMA-Adapter	LLaMA 7B	CLIP ViT-L/14	7.2B	1.2M
mPLUG-Owl	LLaMA 7B	CLIP ViT-L/14	7.2B	0.4B
KOSMOS-2	Decoder Only 1.3B	CLIP ViT-L/14	1.6B	1.6B
Shikra	LLaMA 7B	CLIP ViT-L/14	7.2B	6.7B
μ-G$_2$PT	LLaMA 7B	ViT-G	7B	8B

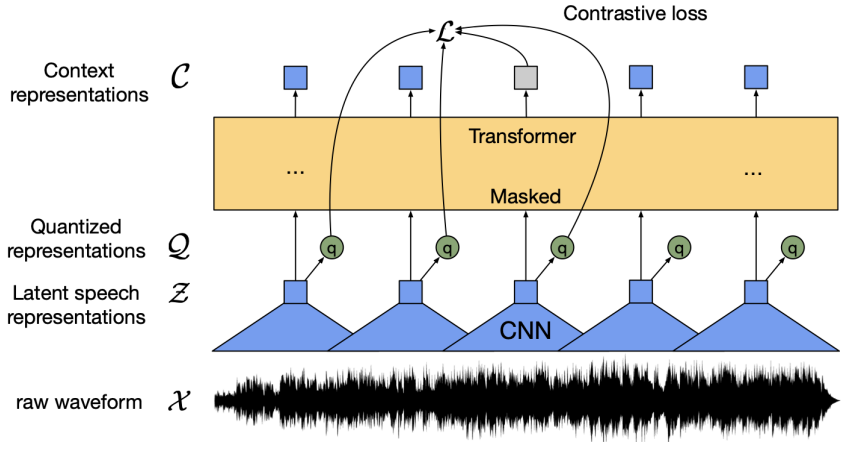
Wav2Vec 2.0, 2020 (Facebook)

- Кодировщик Transformer предобучается в стиле BERT на неразмеченных аудиозаписях и дообучается на размеченных
- **Вход:**
  1. центрированный сырой сигнал WAV
  2. temporal CNN + LayerNorm + GeLU на нем для «токенизации» фреймов
  3. CNN + GeLU на выходах 2 для получения векторов позиций
  4. выходы 2 и 3 складываются и нормализуется
- В качестве unsupervised target используется квантизация векторов входа
- **Квантизованное представление:**
  1. заводится G словарей обучаемых кодов-векторов, по V кодов каждом
  2. для входного вектора диффиринцируемо выбирается лучший код из каждого словаря (с помощью Gumbel-softmax)
  3. G векторов конкатенируются и проецируются в размерность входа
  4. получается квантизованное представление исходного вектора
  5. векторы-коды учатся вместе с моделью

Wav2Vec 2.0, 2020 (Facebook)

- **Предобучение:**
  - N спанов входа по M векторов с перекрытиями маскируются
  - для маскированных входов генерируется квантизованные векторы Q
  - модель учится генерировать для центрального токена спана вектор, который ближе к его вектору из Q, чем к векторам K других маскированных токенов (дистракторов)
  - регуляризатор — максимизация энтропии в распределениях на векторах-кодах (все коды должны участвовать)
- **Дообучение:**
  - квантизация убирается, добавляется полносвязный слой поверх выходов кодировщика
  - на выходах этого слоя идёт обучение на размеченных аудиозаписях

Wav2Vec 2.0, 2020 (Facebook)



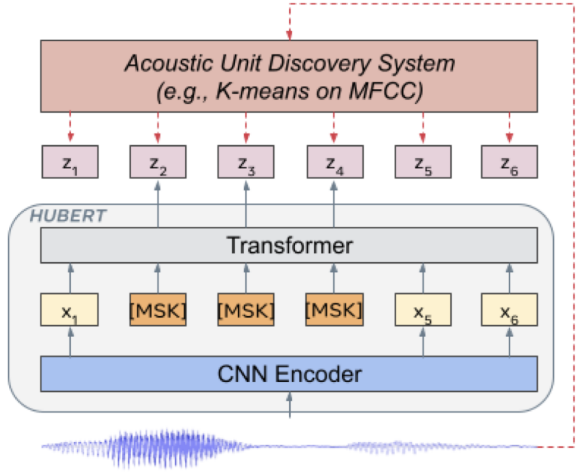
- С небольшими потерями качества удалось уменьшить объём размеченных данных на этапе дообучения в 10-1000 раз

HuBERT, 2021 (Facebook)

- **Идея** та же, что и в Wav2Vec 2.0: учить модель для маскированных фреймов предсказывать альтернативные представления
- Получаются эти представления, — векторы-коды, — с помощью K-means
- **Архитектура**:
  1. CNN-кодировщик аудиосигнала
  2. кодировщик Transformer на его выходах
  3. слой векторов-кодов
  4. слой проекции выходов кодировщика в размерность кодов
- При обучении чередуются два этапа:
  - учится модель K-means на векторах MFCC-признаков фреймов исходного аудиозаписи в WAV
  - учится основная модель-кодировщик

HuBERT, 2021 (Facebook)

- **Обучение кодировщика**:
  - фреймы кодируются CNN и маскируются как в Wav2Vec 2.0
  - вход подаётся в модель, выход проецируется в размерность векторов-кодов
  - между выходным вектором каждого маскированного фрейма и всеми векторами-кодами считается косинусное расстояние
  - результат идёт в softmax, на выходном распределении считается CE-loss



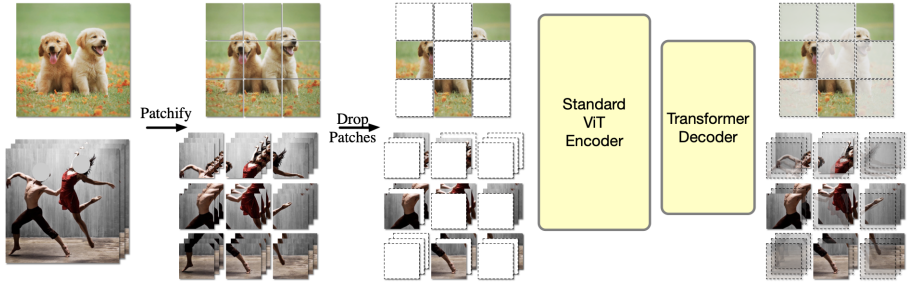
HuBERT, 2021 (Facebook)

- Этапы повторяются итеративно, начиная со 2-й итерации K-means учится на представлениях с промежточных слоёв текущей версии кодировщика, а не на векторах MFCC-признаков
- Для повышения качества вместо одного K-means учатся несколько с разным числом кластеров
- Разбиение на два этапа упрощает и стабилизирует обучение, больше не требуется сложный loss, как в Wav2Vec 2.0
- Предобученная модель дообучается (с замороженной CNN и без K-means) на размеченных записях

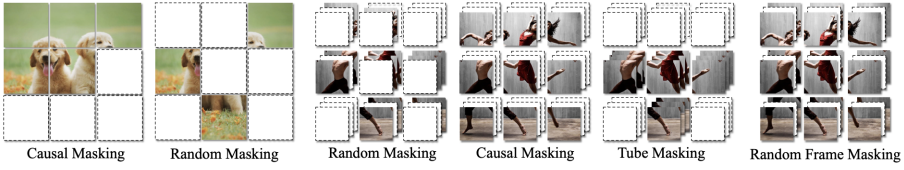
OmniMAE, 2022 (Meta)

- Единая модель для изображений и видео на основе ViT
- Объекты рассматриваются как 4D тензоры ($T \times H \times W \times 3$), у изображений размерность времени всегда 1
- Каждый объект разбивается на N квадратных патчей, из которых M маскируются
- N - M патчей со своими позициями передаются в кодировщик ViT, на выходе векторы
- Эти векторы дополняются M векторами маски, и итоговый набор из N векторов с позициями идёт в декодировщик
- Задача декодировщика — предсказать все пиксели каждого изображения/кадра (MSE на нормализованных значениях пикселя)
- Для ускорения обучения каждый объект обрабатывается при обучении несколько раз с разными масками

OmniMAE, 2022 (Meta)

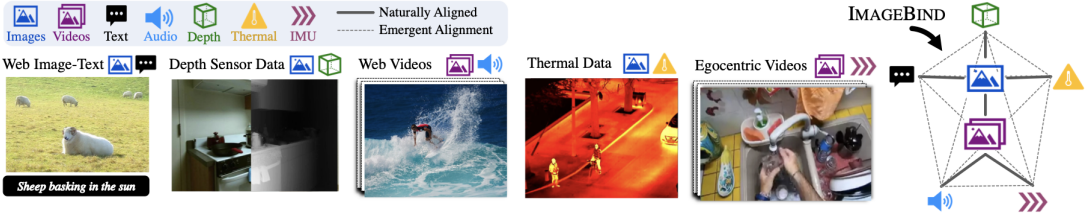


- Ставились эксперименты с разными масками, в итоге для обеих модальностей использовалась Random



ImageBind, 2023 (Meta)

- **Идея**: погрузить 6 модальностей в единое векторное пространство
  - текст
  - изображения
  - аудио
  - видео
  - карты глубины
  - карты температуры
  - замеры гиростабилизатора (IMU)
- **Проблема**: малореалистично собрать попарные данные между всеми модальностями
- **Решение**: использовать изображения в качестве связующего звена между разными модальностями

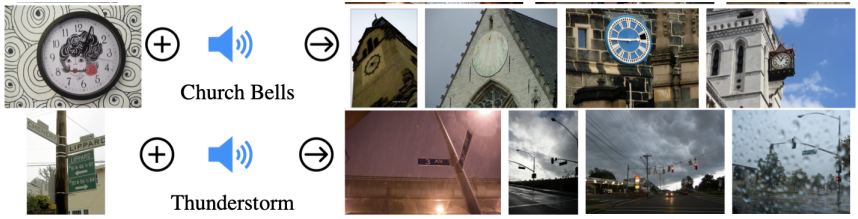


ImageBind, 2023 (Meta)

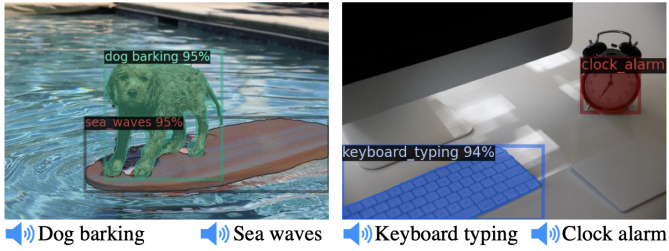
- Пары вида «изображение»-«объект» для объектов 6 модальностей, кодируются и обучаются на InfoNCE loss (верная пара против прочих в батче)
- Кодирование объектов:
  - изображения — ViT (заморожен)
  - видео (2 фрейма) — ViT (заморожен)
  - аудио (2 секунды) — мел-спектрограммы + ViT
  - карты глубины (изображения) — ViT
  - карты температуры (изображения) — ViT
  - IMU — 1D свёртка + кодировщик Transformer
  - текст — CLIP (заморожен)
- У видео и изображений один кодировщик, у всех остальных — свои, у каждого на выходе обучаемая проекция в единую размерность + параметр «температуры» модальности
- Для части пар модальностей есть готовые датасеты, часть данных (для специфических модальностей) собиралась в рамках работы

ImageBind, 2023 (Meta)

- В итоговом пространстве можно складывать векторы разных модальностей:



- Внешней модели (сегментации) с CLIP на входе можно подсунуть вектор аудио:

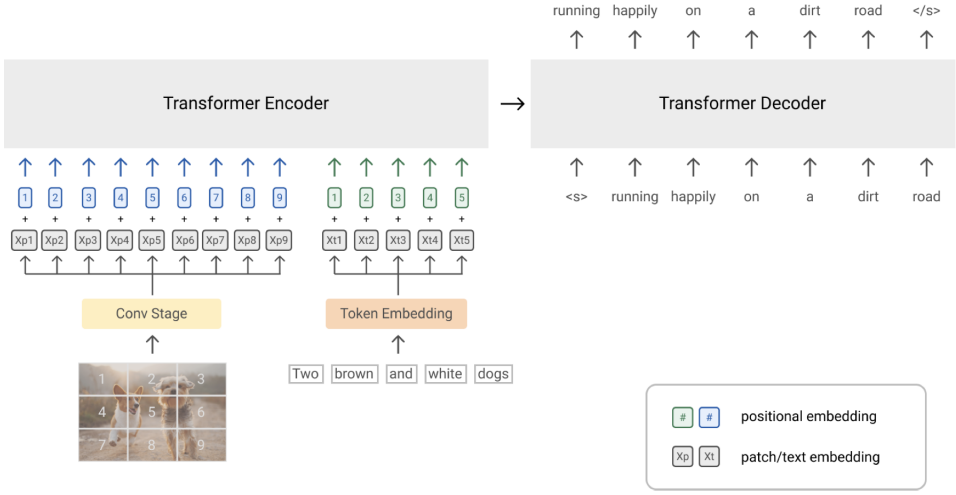


Этап 3

SimVLM, 2021 (Google)

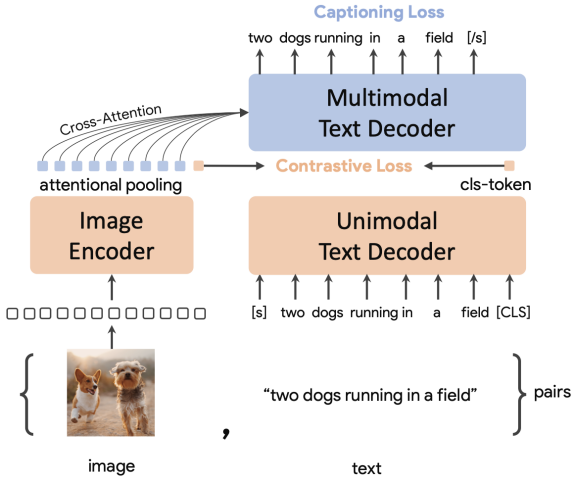
- Полный Transformer, на входе кодировщика изображение и префикс текста, на выходе декодировщика предсказывается суффикс
- Изображение векторизуется с помощью CNN (три первых блока ResNet), каждому патчу на выходе соответствует вектор
- У токенов изображения и текста на входе свои обучаемые позиционные эмбеддинги
- Ко всем токенам вместе применяется обычный self-attention
- У токенов изображений дополнительно есть относительный 2D attention в блоке кодировщика
- Предобучение авторегрессионное одновременно на текстах и парах «текст»-«описание»
- Дообучение на 6 задач с настройкой всех параметров модели

SimVLM, 2021 (Google)



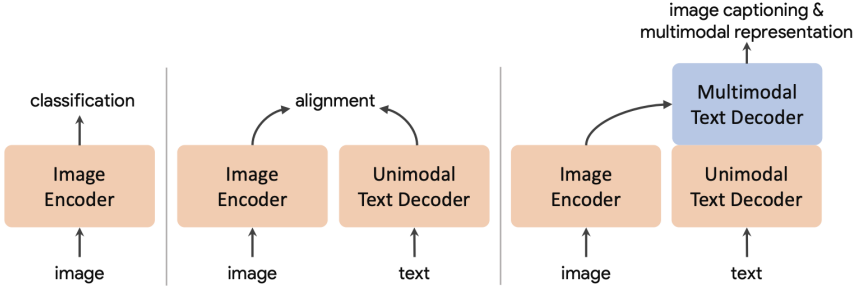
CoCa, 2022 (Google)

- Кодировщик для изображений (ViT или CNN) + общий декодировщик
- На выходе кодировщика векторы патчей изображения и агрегированный с помощью attention pooling общий вектор
- Декодировщик разделён на две части по вертикали
- Первая половина получает на вход только текст префикса и обрабатывает его как обычно, выдавая векторы обычных токенов и вектор CLS-токена



CoCa, 2022 (Google)

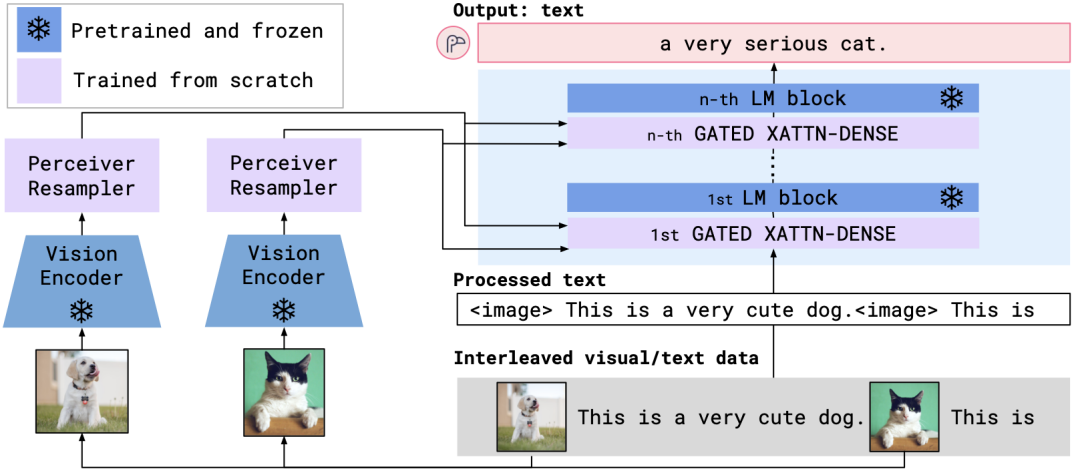
- Векторы текста и изображения объединяются во второй части декодировщика через cross-attention
- Используются два лосса с весами 1 и 2 соответственно:
  - близость между общим вектором изображения и вектором CLS-токена для пары «изображение»-«описание»
  - авторегрессионный лосс на выходе всего декодировщика
- При дообучении можно замораживать кодировщик изображения и доучивать только attention pooling



Flamingo, 2022 (DeepMind)

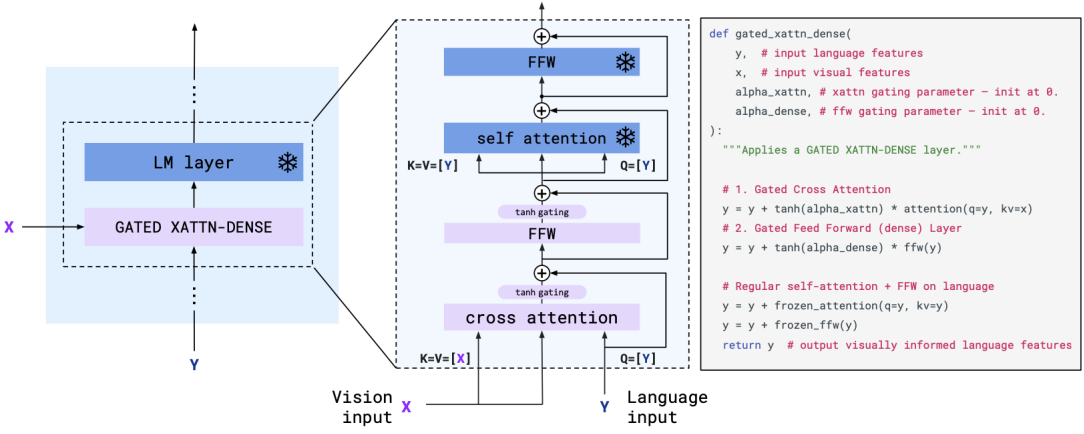
- В основе предобученные кодировщик изображений и LLM (Chinchilla), они замораживаются и дополняются обучаемыми параметрами
- Формат входа: [<img>, ...] text [<img>, ...] text ...
- Кодировщик изображений — NormalizerFree ResNet:
  - 2D изображение $\Rightarrow$ 1D вектор
  - видео $\Rightarrow$ фреймы $\Rightarrow$ 2D + обучаемый вектор метки времени $\Rightarrow$ 1D вектор
- На выходах кодировщика изображения — Perceiver Resampler:
  - преобразует набор векторов любой длины в 64 вектора
  - векторы изображения подаются в cross-attention, формируя ключи и значения
  - запросы — из обучаемых 64 латентных векторов
  - результат проходит через FF-слои и идёт в LLM

Flamingo, 2022 (DeepMind)



Flamingo, 2022 (DeepMind)

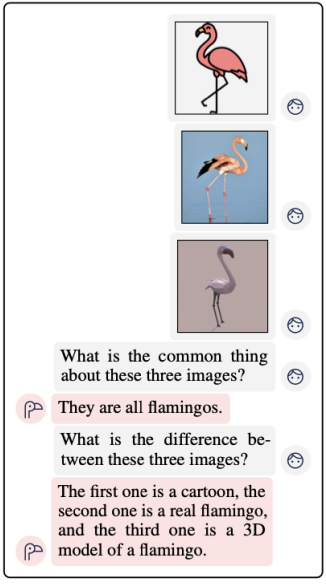
- В LLM между замороженными слоями добавляются новые блоки
- tanh-гейт регулирует пропускную способность, она стартует с 0 и растёт в процессе обучения




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

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

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

Flamingo, 2022 (DeepMind)

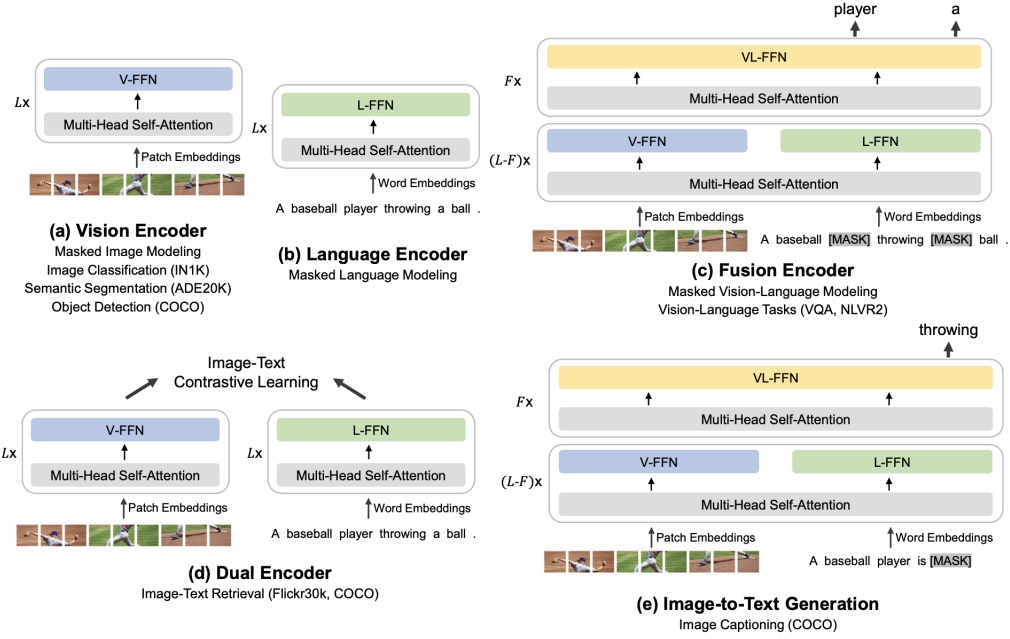
- **Особенности внимания**:
  - токен текста в cross-attention может взаимодействовать только с токенами текста до него и последнего изображения
  - это делается для устранения зависимости от числа изображение на входе
  - взаимодействие с прочими изображениями идёт через self-attention LLM
- В разных версиях модели обучаемые слои добавляются в LLM после каждого 1-го, каждого 4-го и каждого 7-го замороженного слоя
- Модель учится по входу из изображений и текста предсказывать текстовый суффикс (Prefix LM)



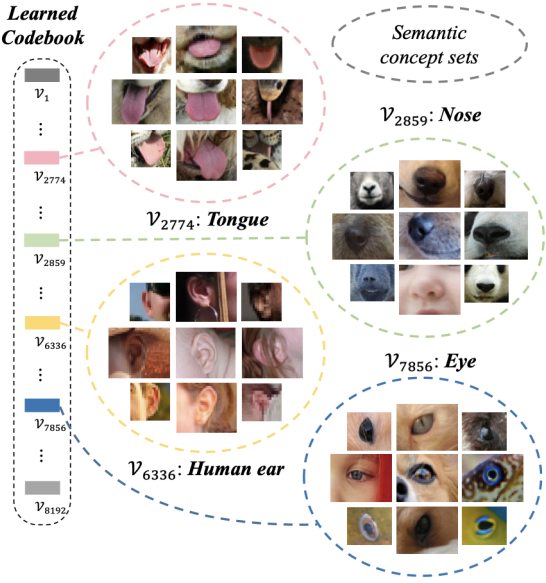
BEiT-3, 2022 (Microsoft)

- Кодировщик Transformer с MoE (Mixture-of-Experts)
- **Архитектура**:
  - на входе векторы патчей изображения и векторы токенов текста
  - первые блоки кодировщика содержат общий self-attention и отдельные FF-слои для каждой модальности
  - в трёх последних блоках и self-attention, и FF-слои общие
- Кодирование патчей как в BEiT v2: ViT + квантизация (VQ), вектор из ViT заменяется на ближайший из обучаемого набора векторов-кодов
- Предобучение на Masked Data Modeling (аналог MLM) на текстах, изображениях и их парах
- Аугментации изображений: обрезка, растяжение, изменения цветов

BEiT-3, 2022 (Microsoft)



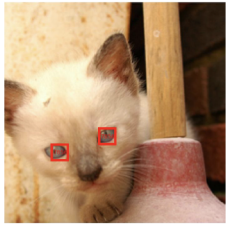
BEiT-3, 2022 (Microsoft)



v_{2774}: *Tongue*

v_{2859}: *Nose*

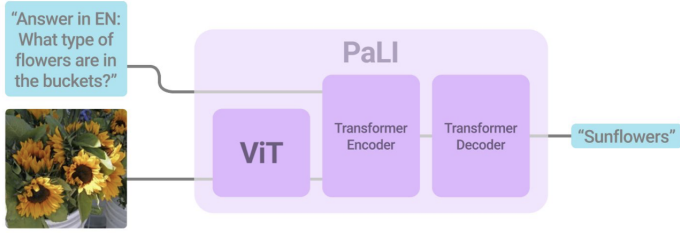
v_{6336}: *Human ear*



v_{7856}: *Eye*

PaLI, 2022 (Google)

- ViT (предобученный и замороженный) + mT5 (обучаемый) с cross-attention на выходы ViT
- **Задачи обучения (если возможно — кросс-язычные)**:
  - MLM, предсказание спанов, замаскированных одним токеном (как в T5)
  - 2 задачи дополнения и генерации alt-текста (текстовый image placeholder)
  - OCR генерация текста по изображению
  - генерация по изображению и тексту вопроса ответа на вопрос (VQA)
  - определение наличия объектов на изображении и генерация списка объектов (только для английского)
  - определение наличия объекта в указанных границах на изображении

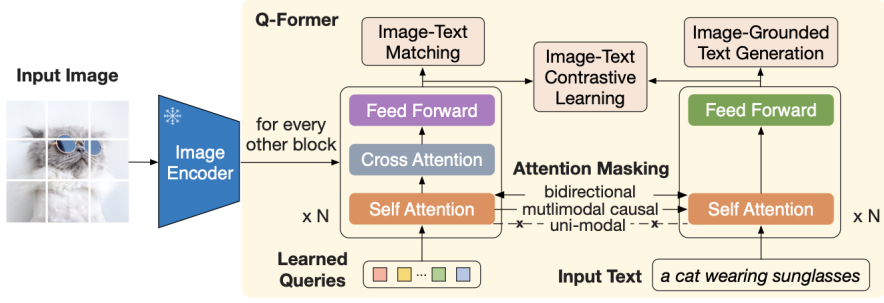


Kosmos-1, 2023 (Microsoft)

- CLIP-ViT для изображений + декодировщик Transformer (Magneto, xPos)
- В предобученном CLIP замораживаются все слои, кроме последнего, для фиксации числа токенов изображения встроен Resampler из Flamingo
- Формат входа: токены текстов и изображений, ограниченные спецтокенами, число и порядок текстов/изображений произвольные
- **Данные**: тексты, изображения, пары «текст»-«изображение», тексты со вставками изображений
- Учится предсказывать по мультимодальному входу текстовый суффикс
- В обучении есть чисто текстовая инструктивная часть (120K примеров)

BLIP-2, 2023 (Salesforce)

- Замороженные кодировщик изображения (CLIP-ViT) и LLM (OPT или Flan-T5) + обучаемый блок между ними (Q-former)
- Q-former извлекает из изображения фиксированное количество векторов (определяется числом обучаемых векторов-запросов Q), т.е. «переводит» изображение в понятные LLM токены
- В основе Q-former — BERT, на входе векторы Q и векторы текста T
- Векторы изображения участвуют в cross-attention с Q self-attention

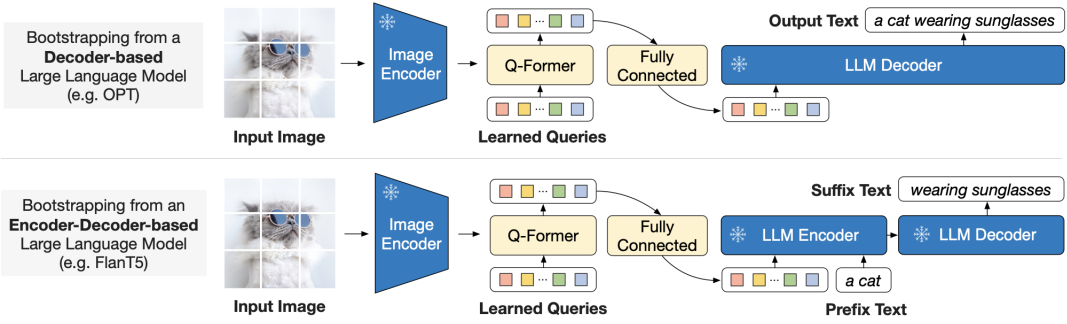


BLIP-2, 2023 (Salesforce)

- Предобучение идёт в 2 этапа (учится только Q-former):
  - кодировщик изображения + Q-former
  - кодировщик изображения + Q-former + LLM
- Задачи первого этапа:
  - Близость между парой «текст»-«изображение»
    - Q и T кодируются независимо
    - для всех пар Q и T в батче на выходе считается близость между всеми Q и вектором CLS-токена T, берётся лучшее значение
    - у верных пар близость должна быть лучшей в батче
  - Генерация текста по изображению:
    - векторы Q могут в self-attention смотреть только на себя
    - векторы T могут смотреть на все векторы Q и на всю T до себя
  - Сопоставление изображения и текста
    - все векторы в self-attention могут смотреть на всех
    - выходные представления Q отдельно подаются в линейный классификатор, логиты усредняются и дают значение качества сопоставления
    - для верной пары значение должно быть высоким, для прочих — ниже

BLIP-2, 2023 (Salesforce)

- Итоговые Q проецируются обучаемым полносвязным слоем в размерность LLM
- Они становятся входом LLM и могут быть дополнены токенами текста
- На втором этапе полная модель с GPT-like [T5-like] LLM обучается предсказывать суффикс по изображению [и тексту префикса]



LLaVA, 2023 (University of Wisconsin–Madison)

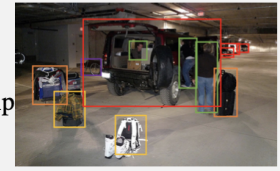
- Предобученный CLIP-ViT с обучаемым проекционным слоем + LLaMA
- Основной упор не на модель, а на подготовку инструкционного датасета
  - пар «изображение»-«текст» много, а инструкций по ним — мало
  - генерировать вручную долго и дорого, можно использовать GPT-4
- Простой поход по генерации одношагового примера по изображению:
  - вход: случайная формулировка запроса на описание + изображение
  - выход: описание изображения
- Продвинутый подход без использования изображения:
  - набор данных COCO с изображениями, сегментацией и описаниями
  - промпты настраивают GPT-4 по описанию изображения и сегментации сгенерировать 3 типа данных:
    - диалог по содержимому изображения
    - развёрнутое детальное описание изображения
    - сложный вопрос к логике изображения и ответ-рассуждение на него

LLaVA, 2023 (University of Wisconsin–Madison)

```
messages = [ {"role":"system", "content": f"""You are an AI visual assistant, and you are seeing a single image. What you see are provided with five sentences, describing the same image you are looking at. Answer all questions as you are seeing the image.

Design a conversation between you and a person asking about this photo. The answers should be in a tone that a visual AI assistant is seeing the image and answering the question. Ask diverse questions and give corresponding answers.

Include questions asking about the visual content of the image, including the object types, counting the objects, object actions, object locations, relative positions between objects, etc. Only include questions that have definite answers:
(1) one can see the content in the image that the question asks about and can answer confidently;
(2) one can determine confidently from the image that it is not in the image. Do not ask any question that cannot be answered confidently.

Also include complex questions that are relevant to the content in the image, for example, asking about background knowledge of the objects in the image, asking to discuss about events happening in the image, etc. Again, do not ask about uncertain details. Provide detailed answers when answering complex questions. For example, give detailed examples or reasoning steps to make the content more convincing and well-organized. You can include multiple paragraphs if necessary."""}
]
```

LLaVA, 2023 (University of Wisconsin–Madison)

Context type 1: Captions



A group of people standing outside of a black vehicle with various luggage.
Luggage surrounds a vehicle in an underground parking area
People try to fit all of their luggage in an SUV.
The sport utility vehicle is parked in the public garage, being packed for a trip
Some people with luggage near a van that is transporting it.

Context type 2: Boxes

person: [0.681, 0.242, 0.774, 0.694], person: [0.63, 0.222, 0.686, 0.516], person: [0.444, 0.233, 0.487, 0.34], backpack: [0.384, 0.696, 0.485, 0.914], backpack: [0.755, 0.413, 0.846, 0.692], suitcase: [0.758, 0.413, 0.845, 0.69], suitcase: [0.1, 0.497, 0.173, 0.579], bicycle: [0.282, 0.363, 0.327, 0.442], car: [0.786, 0.25, 0.848, 0.322], car: [0.783, 0.27, 0.827, 0.335], car: [0.86, 0.254, 0.891, 0.3], car: [0.261, 0.101, 0.787, 0.626]

Response type 1: conversation

Question: What type of vehicle is featured in the image?
Answer: The image features a black sport utility vehicle (SUV).
Question: Where is the vehicle parked?
Answer: The vehicle is parked in an underground parking area, likely in a public garage.
Question: What are the people in the image doing?
Answer: The people are trying to fit all of their luggage into the SUV, likely preparing for a trip.

LLaVA, 2023 (University of Wisconsin–Madison)

Предобучение:

- 600K пар конвертируются в одношаговые диалоги простым методом
- И CLIP, и LLM заморожены, обучается только проекционный слой для сопоставления векторов изображения пространству эмбеддингов LLaMA

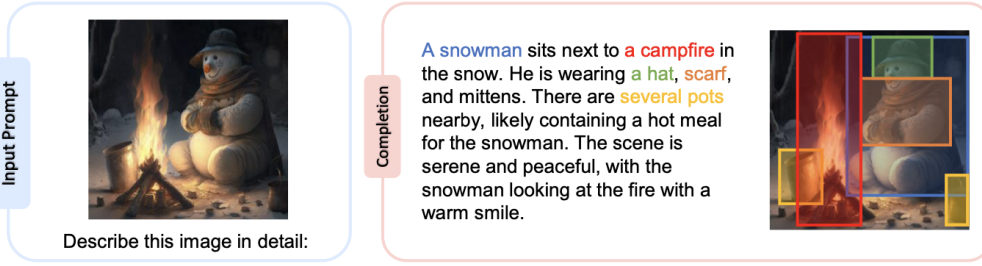
Дообучение:

- 160K инструкционных примеров получаются продвинутым методом
- CLIP заморожена, учатся проекционный слой и LLM

Visual input example, Chicken Nugget Map:	
	Sometimes I just look at pictures of the earth from space and I marvel at how beautiful it all is. 
User LLaVA	Can you explain this meme in detail? The meme in the image is a creative and humorous take on food, with a focus on chicken nuggets, as the center of the universe. The meme begins with the phrase "Sometimes I just look at pictures of the Earth from space and I marvel at how beautiful it all is..." The main image features a tray of cooked chicken nuggets, with the chicken pieces positioned to resemble the continents and islands on a world map. The meme playfully suggests that the chicken nuggets represent the Earth, and the various locations depicted in the photo are actually chicken nugget versions of different places. The punchline of the meme appears in smaller text at the bottom, reading "I mean, it's not the real Earth, but how beautiful it is all is." This lighthearted and entertaining meme combines food and humor, using the image of chicken nuggets to create a funny and imaginative representation of the world.
User GPT-4 [32]	Can you explain this meme? This meme is a joke that combines two unrelated things: pictures of the earth from space and chicken nuggets. The text of the meme suggests that the image below is a beautiful picture of the earth from space. However, the image is actually of chicken nuggets arranged to vaguely resemble a map of the world. The humor in this meme comes from the unexpected juxtaposition of the text and the image. The text sets up an expectation of a majestic image of the earth, but the image is actually something mundane and silly.
User BLIP-2	Can you explain this meme in detail? sometimes i just look at pictures of the earth from space and marvel how beautiful it is
User OpenFlamingo	Can you explain this meme in detail? It's a picture of a chicken nugget on the International Space Station.

Kosmos-2, 2023 (Microsoft)

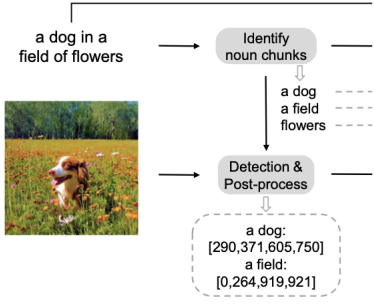
- Архитектура та же, что и Kosmos-1 (CLIP-ViT + декодировщик)
- Особенности:
  - возможность модели выделять запрошенные объекты на изображениях
  - возможность модели фокусироваться на объектах на изображении, которые выделил пользователь



- Ключевой результат — новый набор данных GrIT (**Gr**ounded **I**mage-**T**ext)

Kosmos-2, 2023 (Microsoft)

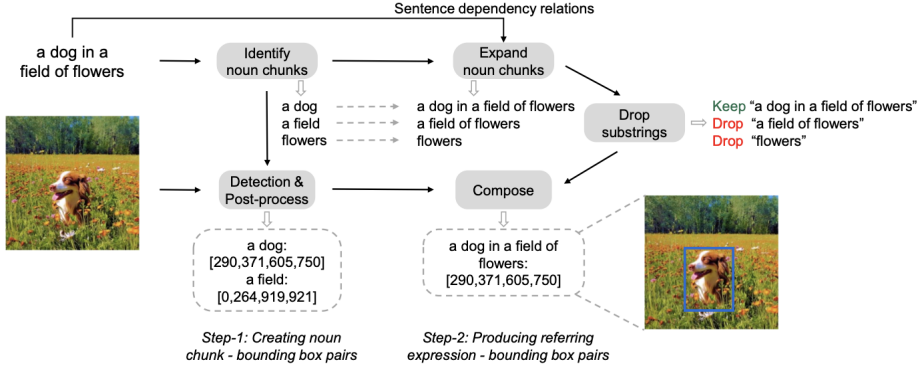
- **Сбор датасета**:
  - выделение существительных (SpaCy), удаление абстрактных понятий
  - подача результатов и изображения в предобученную модель, выделяющую объекты (GLIP)
  - выделение GLIP границ объектов, фильтрация результатов (уверенность модели, доля пересечения с другими), исключение примеров без объектов



Step-1: Creating noun chunk - bounding box pairs

Kosmos-2, 2023 (Microsoft)

- **Сбор датасета**:
  - генерация SpaCy дерева синтаксических зависимостей текста описания
  - расширение существительных их поддеревьями до выражений, фильтрация результатов (полное вхождение в другое выражение)
  - **пример** = выражение + изображение + границы объекта
- **Датасет**: 91M изображений, 137M границ объектов, 115M выражений



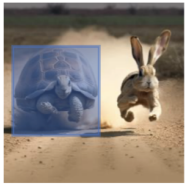
Kosmos-2, 2023 (Microsoft)

- **Дискретизация границ объектов:**
  - изображение фиксированного разрешения делится на маленькие патчи
  - патч характеризуется координатами своего центра на изображении
  - для каждого патча в словарь модели добавляются токены
  - граница объекта = координаты левого верхнего и правого нижнего углов
  - LLM получает границу на вход в виде токенов патчей этих углов
- **Формат входа**:
  - object-text <box><left-up-token><right-down-token></box>
  - на случай нескольких границ для одного объекта есть токен <delim>
  - во всём остальном модель повторяет Kosmos-1

<s> <image> Image Embedding </image> <grounding> It <box><loc$_{44}$><loc$_{863}$></box> seats next to a campfire <box><loc$_{4}$><loc$_{1007}$></box> </s>

Kosmos-2, 2023 (Microsoft)

- Модель основана на Kosmos-1, новые данные включены в старый датасет
- SFT проводился на данных LLaVA, Unnatural Instructions и FLAN v2 + дополнительные задачи на данных GrIT

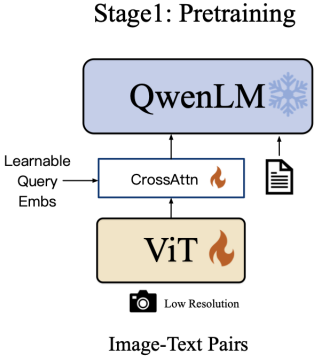
Input Prompt	 Question: Why is this animal unusual? Answer:	 Question: What is it? Answer:	 Question: What is the biggest difference betwen bottle-1 and bottle-2? Answer:
Completion	This animal is unique because it is a turtle. Turtles are known for their slow and steady pace, whereas rabbits are known to be agile and quick. It is not common for a turtle to be faster than a rabbit, especially in terms of speed.	a buoy attached to the boat.	The biggest difference between bottle-1 and bottle-2 is the label.

Qwen-VL, 2023 (Alibaba)

- Адаптация Qwen-7B к изображениям с помощью Openclip ViT
- Модель тоже умеет работать с границами объектов
- Изображения переводятся в 256 векторов через cross-attention с латентными векторами-ключами (как в Perceiver Resampler Flamingo)
- В cross-attention используется дополнительное 2D абсолютное позиционное кодирование
- **Формат входа**:
  - изображение отделяется спецтокенами `<img>` и `</img>`
  - координаты точек границ объектов (слева сверху и справа снизу) нормируются в [0, 1000) и подаются в виде строки «(A, B), (C, D)»
  - Для её выделения используются токены `<box>` и `</box>`
  - Текст, описывающий объект в границах, отделяется `<ref>` и `</ref>`

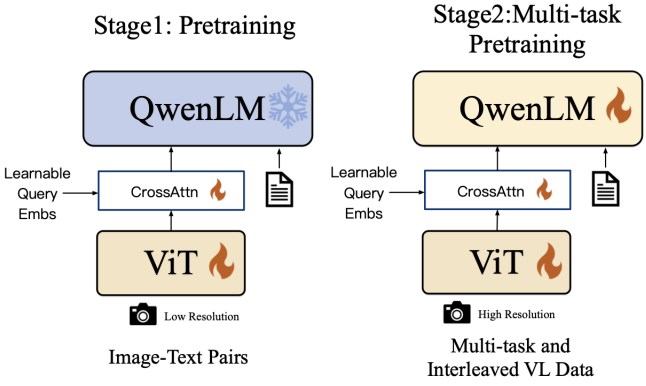
Qwen-VL, 2023 (Alibaba)

- **1-й шаг предобучения**:
  - на 1.4B парах «изображение»-«текст»
  - разрешение изображений 224×224
  - LLM заморожена, учатся ViT и адаптер



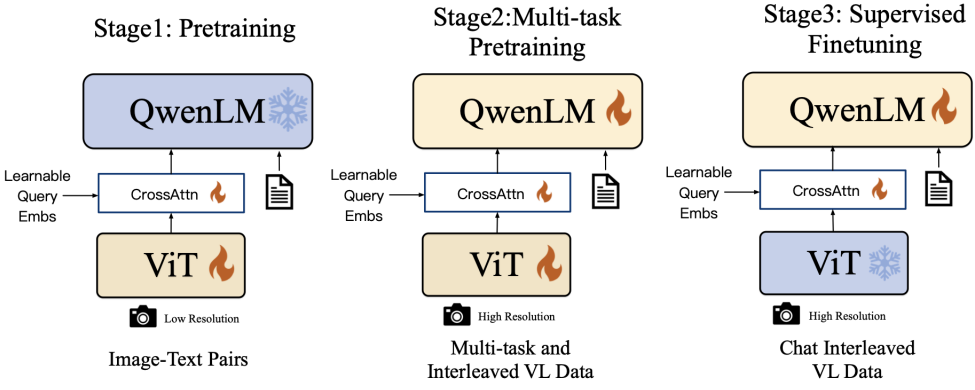
Qwen-VL, 2023 (Alibaba)

- **2-й шаг предобучения**:
  - на >75M примерах из 20 наборов данных для 7 задач
  - разрешение изображений 448×448
  - Учится вся модель



Qwen-VL, 2023 (Alibaba)

- **SFT**:
  - 350K инструкций, собранных разными способами
  - разрешение изображений 448 $\times$ 448
  - ViT заморожен, учатся LLM и адаптер



Qwen-VL, 2023 (Alibaba)

```c
#include <stdlib.h>

int solution(int *A, int A_length) {
    int ans = 0;
    for (int i = 1; i < A_length; i++) {
        if (A[i] < ans) {
            ans = A[i];
        }
    }
    return ans;
}
```

The solution in the image is to find the minimum value in an array. And there is a bug in the function. Fix the bug.

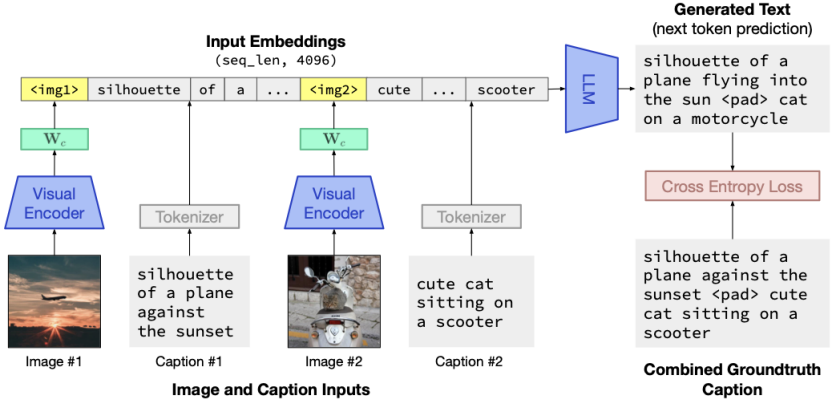
```c
int solution(int A[], int n) {
  int ans = A[0];
  for (int i = 1; i < n; i++) {
    if (A[i] < ans)
      ans = a[i];
  }
  return ans;
}
```

Can you find spider man and Hulk?

Этап 4

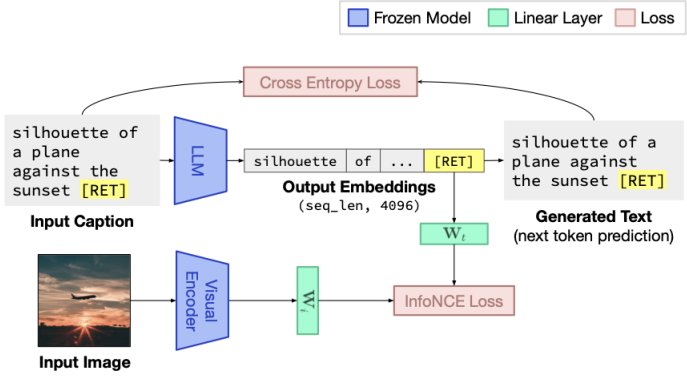
FROMAGe, 2023 (Carnegie Mellon University)

- Замороженные CLIP-ViT и OPT + обучаемые проекции и доп. токен
- **Задача 1 (генерация описаний)**:
  - выходной вектор CLIP проецируется в векторы «токенов» (reshape большого выхода линейного слоя проекции на k элементов)
  - последовательность — набор нескольких случайных пар: модель учится для токена текста смотреть именно на нужное для него изображение



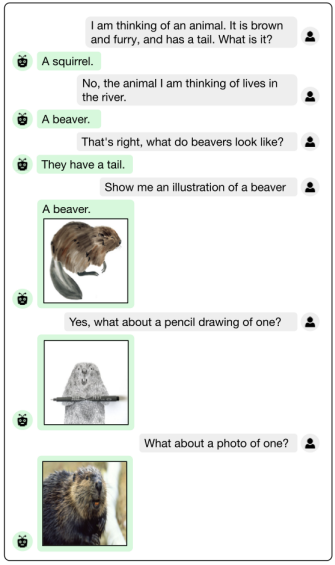
FROMAGe, 2023 (Carnegie Mellon University)

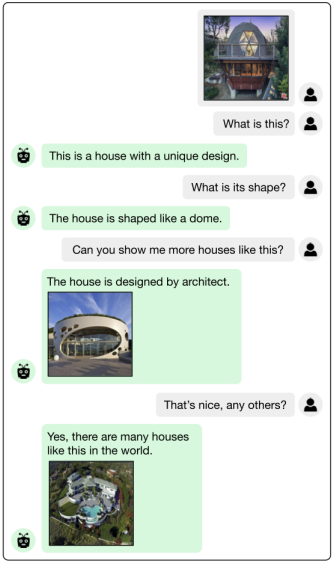
- **Задача 2 (сопоставление)**:
  - Новый спецтокен [RET] в конце текста добавляет шаг self-attention, учитывающий все токены в авторегрессионной модели
  - Выходной вектор [RET] улучшает качество сопоставления
  - Учатся по слою проекции для вектора [RET] и выходного вектора CLIP + эмбеддинг токена [RET]
  - Негативные примеры в loss — все неверные пары батча

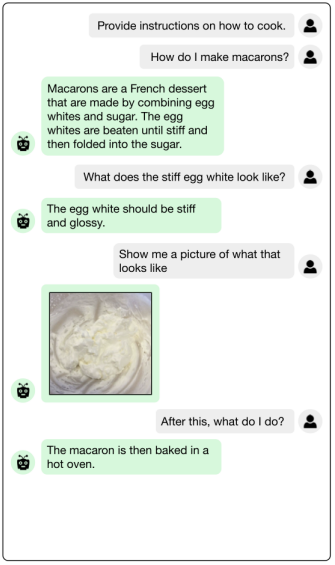


FROMAGe, 2023 (Carnegie Mellon University)

- Мультимодальные (картиночные) ответы модели обеспечиваются поиском по базе, больше контекста — лучше результат

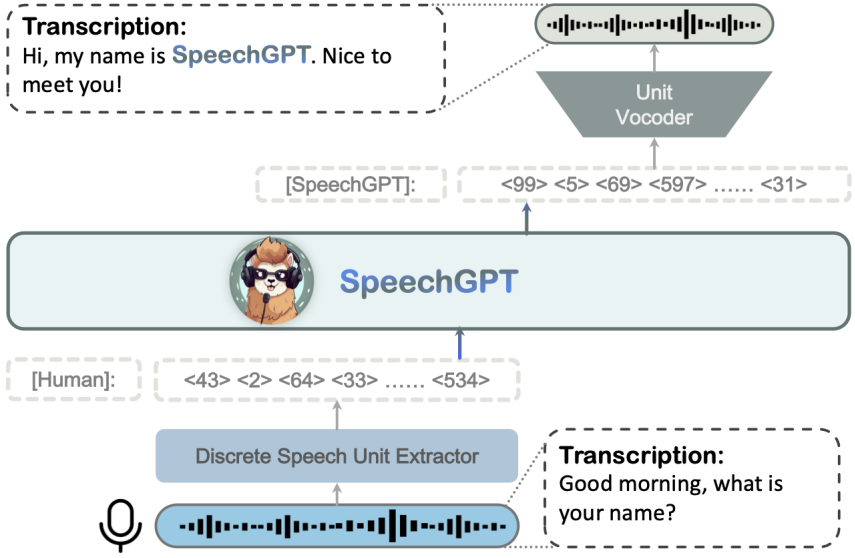






SpeechGPT, 2023 (Fudan University)

- Кодировщик HuBERT (с K-Means) + LLaMA + вокодер HiFi-GAN



SpeechGPT, 2023 (Fudan University)

- Собран собственный набор данных SpeechInstruct из двух частей
- **Cross-modal Instruction**:
  - из открытых датасетов отобраны и дедуплицированы пары «текст»-«аудио» (9M)
  - с помощью GPT-4 сгенерированы 100+ инструкций-запросов на транскрибацию или синтез речи
  - инструкции и данные случайно соединяются в тройки
  - тройки конкатенируются в «диалоги» по максимальной длине входа LLM
- **Chain-of-Modality Instruction**:
  - на Cross-modal обучена вспомогательная модель-кодировщик
  - с её помощью для 38K инструкционных примеров из moss-002-sft-data сгенерированы векторы аудио
  - сформированы 38K четвёрок «(текст входа, аудио входа, текст ответа, аудио ответа)»
  - на их основе формируются Chain-of-Modality инструкции

SpeechGPT, 2023 (Fudan University)

Speech Instruction-Speech Response:
[Human]: This is a speech instruction: {SpeechI}. And your response should be speech. You can do it step by step. You can first transcribe the instruction and get the text Instruction. Then you can think about the instruction and get the text response. Last, you should speak the response aloud <eoh>. **[SpeechGPT]**: **[tq]** {TextI}; **[ta]** {TextR}; **[ua]** {SpeechR}<eoa>.

Speech Instruction-Text Response:
[Human]: This is a speech instruction: {SpeechI}. And your response should be text. You can do it step by step. You can first transcribe the instruction and get the text instruction. Then you can think about the instruction and get the text response. <eoh>. **[SpeechGPT]**: **[tq]** {TextI}; **[ta]** {TextR}<eoa>.

Text Instruction-Speech Response:
[Human]: This is a text instruction: {TextI}. And your response should be speech. You can do it step by step. You can think about the instruction and get the text response. Then you should speak the response aloud <eoh>. **[SpeechGPT]**: **[ta]** {TextR}; **[ua]** {SpeechR}<eoa>.

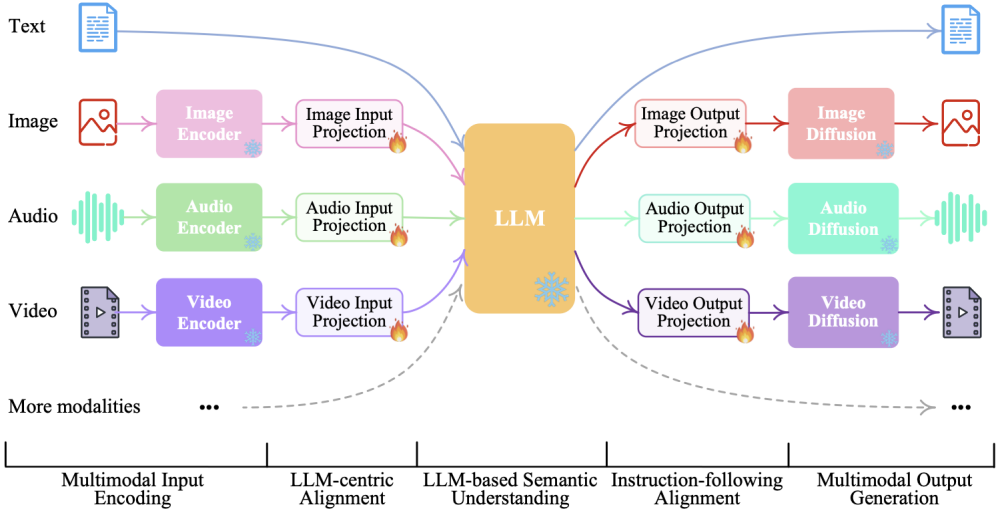
Text Instruction-Text Response:
[Human]: This is a text instruction: {TextI}. And your response should be text. You can think about the instruction and get the text response. **[SpeechGPT]**: **[ta]** {TextR}<eoa>.

SpeechGPT, 2023 (Fudan University)

- **Предобучение-адаптация к модальности**:
  - в LLM добавляются новые токены, соответствующие дискретным аудио-токенам
  - предобучение LLM идёт на предсказание следующего аудио-токена на неразмеченных записях
  - Кодировщик аудио (скорее всего) заморожен
- **Кросс-модальный SFT**: вся модель учится на смеси текстовых и Cross-modal инструкций
- **Chain-of-Modality SFT**: замороженная модель с адаптером LoRA учится на Chain-of-Modality инструкциях
- Обучение идёт на предсказание следующего токена текстового выхода (видимо, после транскрибации результата)

NExT-GPT, 2023 (National University of Singapore)

- Кодировщик ImageBind + LLM Vicuna + диффузионные генераторы (все предобученные и замороженные)



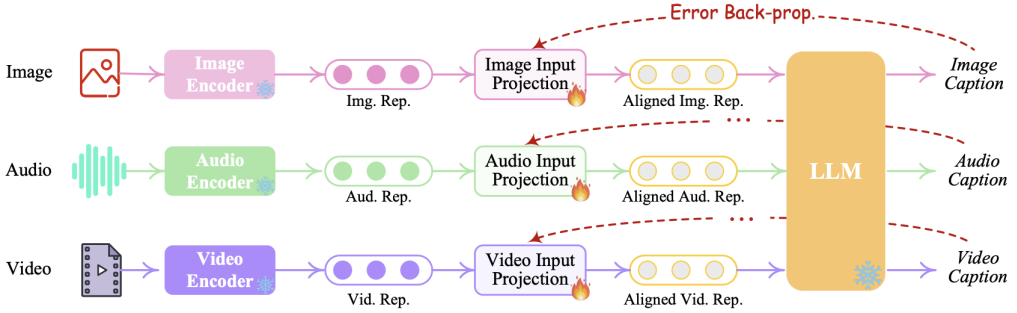
NExT-GPT, 2023 (National University of Singapore)

	Encoder		Input Projection		LLM		Output Projection		Diffusion	
	Name	Param	Name	Param	Name	Param	Name	Param	Name	Param
Text	—	—	—	—			—	—	—	—
Image					Vicuna [12]	7B ❄	Transformer	31M 🔥	SD [68]	1.3B ❄
Audio	ImageBind [25]	1.2B ❄	Linear	4M 🔥	(LoRA	33M 🔥)	Transformer	31M 🔥	AudioLDM [51]	975M ❄
Video							Transformer	32M 🔥	Zeroscope [8]	1.8B ❄

- LLM получает на вход векторы всех входных сущностей + текст
- При генерации LLM решает, нужно ли сгенерировать токен текста или объект какой-то иной модальности
- Во втором случае генерируется один из спецтокенов этой модальности
- После окончания работы LLM этот токен вместе с текстом передаётся в соответствующую диффузионнную модель
- Один ответ модели может использовать несколько разных диффузий

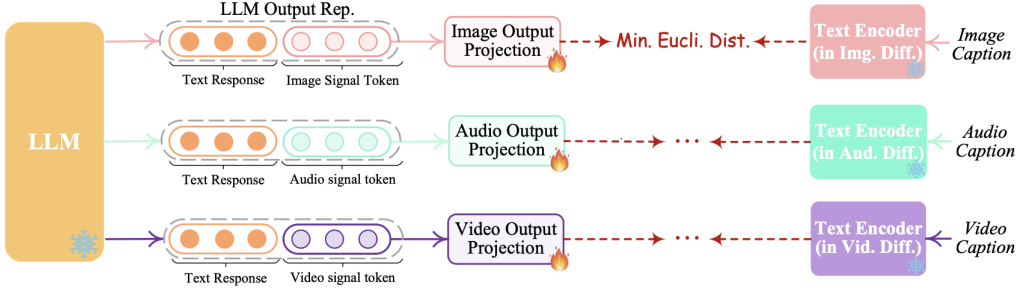
NExT-GPT, 2023 (National University of Singapore)

- Обучение входных линейных проектирующих слоёв идёт на задачу генерации текста описания объекта



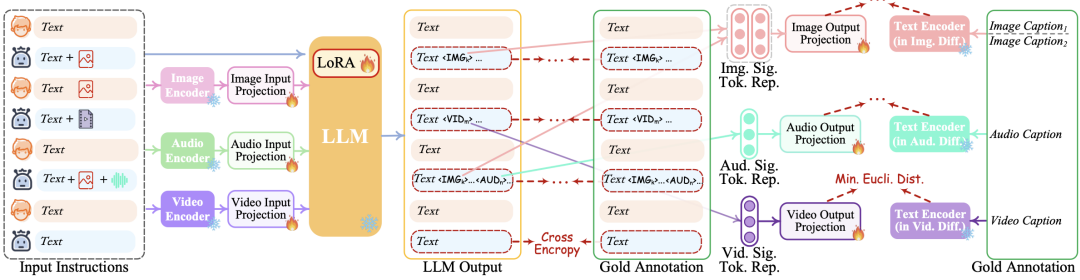
NExT-GPT, 2023 (National University of Singapore)

- Обучать полноценное сопоставление всех диффузионных декодировщиков и LLM долго и дорого
- Диффузии обусловлены на текст описания
- Выходные проектирующие слои учатся приближать представления текста и спецтокенов модальности к представлениям текста в диффузиях

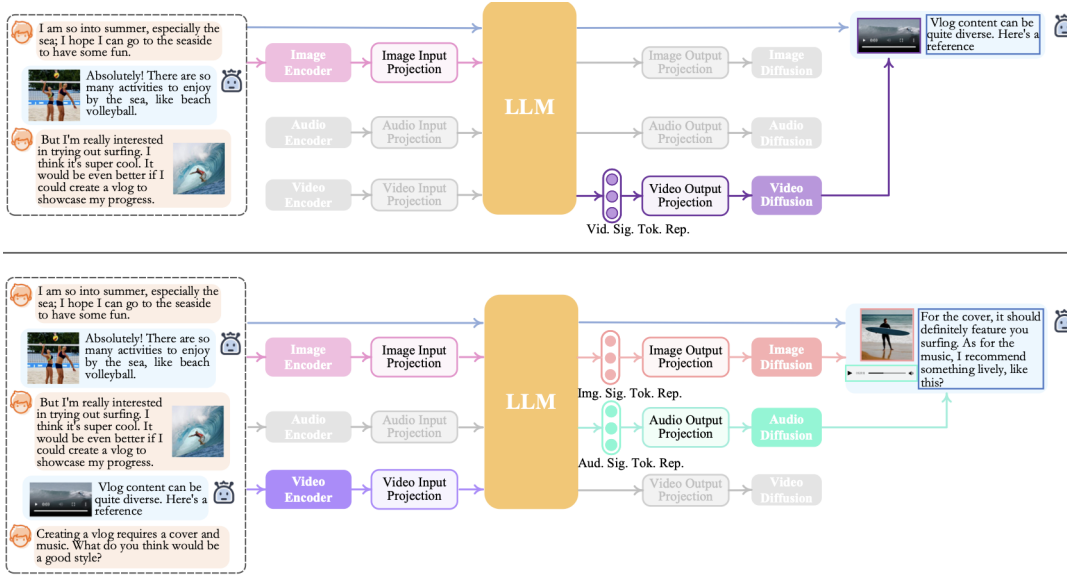


NExT-GPT, 2023 (National University of Singapore)

- В инструктивном дообучении участвует LLM (с адаптером LoRA)
- Модель дообучается на размеченных MM-диалогах, в процессе дополнительно настраиваются выходные проекции
- Обучение на открытых наборах данных + собственный датасет MosIT (**Mo**dality-**s**witching **I**nstruction **T**uning)



NExT-GPT, 2023 (National University of Singapore)



Выводы

- ViT и CLIP оказались простыми и удачными моделями, используемыми повсеместно до сих пор
- Общий тренд на добавление мультимодальности путём комбинирования предобученных моделей
- Дообучение на задачи MM обычно требует настройки только части параметров и/или адаптеров типа линейных проекций и Q-former
- Лучший MM-векторизатор на текущий момент — ImageBind, поверх него делаются свежие MLLM
- LLaVA демонстрирует пример генерации качественного инструктивного MM набора данных
- Kosmos-2 — сильный пример перехода от понимания моделью изображения в целом до взаимодействия с объектами на нём
- С помощью FROMAGe можно относительно просто и качественно привнести в модель понимание и поиск изображений

Полезные ссылки

- MMBench: Is Your Multi-modal Model an All-around Player? (2023)
- An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (2020)
- Learning Transferable Visual Models From Natural Language Supervision (2021)
- wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations (2020)
- HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units (2021)
- OmniMAE: Single Model Masked Pretraining on Images and Videos (2022)
- ImageBind: One Embedding Space To Bind Them All (2023)
- SimVLM: Simple Visual Language Model Pretraining with Weak Supervision (2021)
- CoCa: Contrastive Captioners are Image-Text Foundation Models (2022)
- Flamingo: a Visual Language Model for Few-Shot Learning (2022)
- Image as a Foreign Language: BEiT Pretraining for All Vision and Vision-Language Tasks (2022)

Полезные ссылки

- PaLI: A Jointly-Scaled Multilingual Language-Image Model (2022)
- BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models (2023)
- Language Is Not All You Need: Aligning Perception with Language Models (2023)
- Visual Instruction Tuning (2023)
- Kosmos-2: Grounding Multimodal Large Language Models to the World (2023)
- Qwen-VL: A Frontier Large Vision-Language Model with Versatile Abilities (2023)
- Grounding Language Models to Images for Multimodal Inputs and Outputs (2023)
- SpeechGPT: Empowering Large Language Models with Intrinsic Cross-Modal Conversational Abilities (2023)
- NExT-GPT: Any-to-Any Multimodal LLM (2023)
- Vision Language Transformers: A Survey (2023)
- Зоопарк трансформеров: большой обзор моделей от BERT до Alpaca (2023)

Спасибо за внимание!