**Метод Logit Lens**

Logit Lens используется для оценки промежуточных предсказаний трансформера. После каждого слоя берётся скрытое состояние (hidden state), к нему применяется финальная нормализация слоя (если она есть), затем выходная линейная голова, используемая при обычном предсказании токенов. После применения софтмакса получается распределение вероятностей слов из словаря. Это позволяет узнать, какие токены были бы выбраны моделью, если бы предсказание делалось на этом этапе. Применение финальной головы позволяет понять, насколько на данном этапе модель делает хорошие предсказания; нормализация применяется, так как без неё масштаб будет отличаться от выходного слоя и приведёт к некорректным результатам.

Этот метод показывает, как меняются логиты - от более зашумлённых на ранних слоях к более точным на поздних. Однако скрытые состояния промежуточных слоёв могут не лежать в том же линейном пространстве, что и выходной слой. Поэтому полученные логиты могут быть неточными, что ограничивает применение Logit Lens в архитектурах, где выходная голова обучается под финальный слой. В таких случаях возможна ошибка интерпретации, так как промежуточные слои могут использовать иные представления. Простота алгоритма является одновременно его плюсом и минусом: было показано, что метод не может делать предсказания на хорошем уровне для моделей GPT-Neo, BLOOM, OPT 125M и плохо работает с нестандартными блоками.

**Датасеты**

Для применения мультимодальных моделей возьмём два датасета с изображениями, различающиеся по сложности и содержанию:

CLEVR - синтетический датасет, содержащий сцены с простыми геометрическими объектами. Одно из его свойств — контролируемая структура изображений: каждый объект описывается по цвету, форме, размеру и положению.

COCO (Common Objects in Context) - содержит фотографии реальных сцен из разных мест.

**Модели**

Для анализа было выбрано 2 модели:

1. llava-hf/llava-1.5-7b-hf
2. Qwen/Qwen2-VL-2B-Instruct

Они имеют схожую структуру:

* Визуальный модуль- разбивает изображение на патчи, добавляет позиционный энкодинг, преобразует изображение в эмбеддинги слоями трансформера (CLIPVisionTransformer для LLaVA-OneVision, VisionTransformer для Qwen2-VL)
* Мультимодальная проекция - приводит эмбеддинги изображений к нужной текстовой размерности
* Языковая модель - блок который будет использоваться с помощью метода logit lens
  * LLaMa (0-31)
  * Qwen2-2B (0-23)

**Процесс применения метода Logit Lens**

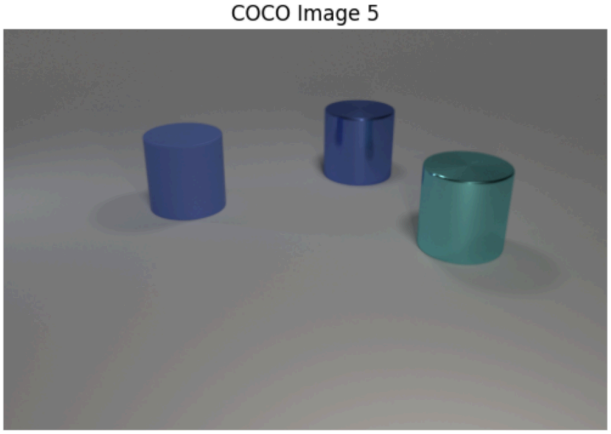
1. Сохранение скрытых состояний после каждого слоя языковой модели.
2. Получение логитов за счёт применения головы. Так как перед головой нормализация не применяется, она не была добавлена в данном случае.
3. Вывод трёх наиболее вероятных токенов. Было решено отобразить несколько токенов, чтобы посмотреть, насколько близкие по смыслу токены предлагает модель.
4. Отображение уверенности модели в токене (вероятности, полученной после применения софтмакса к логитам).

**Анализ логитов**

При первых экспериментах получалось так, что на первых позициях даже поздних слоёв появляются токены, которые не несут смысла (к примеру: in, a, the). Чтобы избежать такой проблемы, было рассмотрено два варианта решения: отслеживать большее количество токенов или видоизменить промпт таким образом, чтобы модель с большей вероятностью выдавала осмысленный токен на первой позиции.
Как оказалось, промпт сильно влияет на результат, поэтому было опробовано несколько вопросов, которые помогают посмотреть, как меняется ответ модели на различные запросы. Они были составлены так, чтобы можно было получить односложный ответ.

```
clevr_prompts = [
    "Color of the main object:",
    "Shape of the main object:",
    "Material of the main object:",
    "Number of main objects:",
]
coco_prompts = [
    "Main object:",
    "What is the person doing?:",
    "Place:",
    "Main animal:",
    "Background shows:"
]
```

Начнем исследование с модели llava-hf/llava-1.5-7b-hf:
1. Вероятность топ-1 токена растёт при продвижении через слои трансформера до 25 слоя, достигает пика на предпоследнем слое и резко падает после последнего. Причина: модель накапливает информацию на более ранних слоях, добавление информации приводит к росту уверенности, далее модель начинает подстраивать ответ и поэтому вероятность не растёт. Последний слой подстраивается под функцию потерь, сглаживает логиты, чтобы учесть важный контекст и скорректировать ответ.



| Layer | Top-1 | Top-2 | Top-3 |
|---|---|---|---|
| Layer 0 | Portail | ensoort | iemann |
| Layer 1 | Portail | archivi | embros |
| Layer 2 | sterd | Архив | idense |
| Layer 3 | sterd | Архив | sierp |
| Layer 4 | sterd | paździer | kwiet |
| Layer 5 | sterd | sierp | paździer |
| Layer 6 | sterd | sierp | paździer |
| Layer 7 | sterd | Censo | quelle |
| Layer 8 | sterd | Censo | penas |
| Layer 9 | bez | penas | gat |
| Layer 10 | □ | none | mannschaft |
| Layer 11 | sterd | othèque | reten |
| Layer 12 | penas | mannschaft | rola |
| Layer 13 | penas | reten | mannschaft |
| Layer 14 | igt | uclide | rola |
| Layer 15 | textt | rola | none |
| Layer 16 | rola | textt | живело |
| Layer 17 | textt | rola | none |
| Layer 18 | None | materials | none |
| Layer 19 | none | None | materials |
| Layer 20 | unknown | gold | Gold |
| Layer 21 | unknown | gold | Gold |
| Layer 22 | materials | material | gold |
| Layer 23 | gold | Gold | golden |
| Layer 24 | gold | Gold | sphere |
| Layer 25 | pl | gold | glass |
| Layer 26 | material | materials | metal |
| Layer 27 | Gold | gold | material |
| Layer 28 | Gold | gold | pl |
| Layer 29 | Gold | gold | Pl |
| Layer 30 | Gold | glass | Pl |
| Layer 31 | G | Pl | Gold |
| Layer 32 | Metal | Pl | Gold |

| Layer | Top-1 | Top-2 | Top-3 |
|---|---|---|---|
| Layer 0 | Portail | ensoort | iemann |
| Layer 1 | Portail | archivi | embros |
| Layer 2 | sterd | Архив | idense |
| Layer 3 | sterd | Архив | sierp |
| Layer 4 | sterd | paździer | Архив |
| Layer 5 | sterd | paździer | sierp |
| Layer 6 | sterd | sierp | paździer |
| Layer 7 | sterd | Censo | penas |
| Layer 8 | bez | arrow | Censo |
| Layer 9 | bez | chan | gat |
| Layer 10 | reten | yman | sterd |
| Layer 11 | reten | sterd | none |
| Layer 12 | Хронологија | penas | reten |
| Layer 13 | reten | None | pf |
| Layer 14 | □ | color | none |
| Layer 15 | none | None | □ |
| Layer 16 | □ | color | Color |
| Layer 17 | □ | color | Color |
| Layer 18 | □ | color | black |
| Layer 19 | green | Green | black |
| Layer 20 | green | Green | green |
| Layer 21 | green | Green | green |
| Layer 22 | green | Green | green |
| Layer 23 | green | Green | green |
| Layer 24 | green | Green | green |
| Layer 25 | green | Green | green |
| Layer 26 | green | Green | green |
| Layer 27 | green | Green | green |
| Layer 28 | Green | green | green |
| Layer 29 | Green | green | Green |
| Layer 30 | Green | green | Blue |
| Layer 31 | Green | Blue | Y |
| Layer 32 | Green | Blue | Silver |

Пример: смотря на топ-3 токена, видно, что в основном слова, относящиеся к теме, появляются примерно на 16-м слое. До этого модель не уверена даже в языке ответа и предлагает разные варианты. Примерно к 18–19-му слою формируется ответ, близкий к правильному, на слоях 27–28 ответ корректируется, и на последнем появляется финальный ответ (иногда он отличается от предыдущих и в итоге оказывается правильным). Такая ситуация часто прослеживается на формах объектов, где на последнем слое модель склоняется в пользу “плоских” форм (*square* вместо *cube*).

2. Можно проследить, в каком порядке модель формирует знание об изображении: модель раньше начинает распознавать цвет, затем переходит к форме и количеству, а после — к материалу. Это объясняется тем, что цвет является более простым признаком, чем форма и материал.
3. Модель путает схожие формы (*cube/square*, *sphere/circle*). Это может объясняться тем, что текстовая модель обучалась на текстах и на последнем слое сглаживает логиты в пользу более частотного варианта. Такой же ответ сохраняется, если задать, что объект — 3D.
4. На несинтетическом датасете прослеживается другая закономерность: модель раньше приходит к правильному ответу, что объясняется тем, что реальные сцены чаще встречаются в обучающей выборке.
5. Модель менее уверена в ответе на реальных данных.

Далее был проведен анализ для модели Qwen/Qwen2-VL-2B-Instruct:

1.Модель выдаёт более точные и специфичные ответы, в отличие от LLaVa.  Например, предлагаются более редкие цвета (cyan), лучше справляется с задачами на подсчёт, лучше различает 3D и 2D постановку. Этот эффект достигнут за счёт лучшего обучения визуальной части модели.

**Проблема мультимодальных моделей:**
**a.Галлюцинации**. Модель выдает в результате объект, которого нет на изображении, или не говорит, что объект отсутствует. Для решения проблемы я предлагаю дообучить модель на данных, где будут встречаться вопросы с ответом “не знаю” или сделать модель более сложной. Этот способ прост в реализации и уменьшает чрезмерную уверенность модели, однако может привести к ухудшению качества. Второе решение - взять более качественный визуальный энкодер. Это делает модель более сложной и умной, но в то же время требует больше вычислительных ресурсов.

В данной статье описываются методы борьбы с галлюцинациями https://arxiv.org/pdf/2404.18930:
1.Включить в датасет отрицательные примеры (подход, предложенный мной)

2. Reasoning Data. Заставлять модель обосновывать ответ, давая ей и правильный и неправильные ответы. Это повышает устойчивость модели к галлюцинациям, но требует сложной адаптации данных.

4. Увеличить разрешение визуального энкодера. Это повышает точность на сложных изображениях, но требует больше ресурсов и не всегда оправданно.

5. Контрастное обучение для сравнения галлюцинаций и адекватных ответов. В этом подходе нужно заранее найти примеры галлюцинаций, что может быть сложной задачей.

**b.Оптимальное объединение**. Поскольку большинство проблем,  с которыми я столкнулась являются галлюцинациями, я хочу затронуть проблему оптимального объединения данных разных модальностей (текстов и картинок). Модели, примененные мной используют для этого cross-attention. Это даёт возможность анализировать структуру модели с помощью методов таких как logit lens, делает модель более гибкой. Однако этот подход может привести к галлюцинациям и большой зависимости от качества визуального энкодера. Для решения данной проблемы предлагается учить изображение и текст в одном пространстве (однако этот подход не всегда оказывается более эффективным)