**Введение**

Основная цель данного исследования — изучить возможности метода Logit Lens при интерпретации больших языковых моделей (Large Language Models, LLM), в том числе мультимодальных вариаций (Vision-Language Models, VLM). В частности, в работе уделяется внимание тому, как изменяется распределение логитов по слоям нейросетевой архитектуры при обработке текстовых и визуальных данных. Актуальность темы обусловлена быстро растущей популярностью интерпретируемых решений в сфере генеративных моделей и необходимостью более глубокого понимания того, как именно крупные модели «думают» при формировании ответов.

Целями исследования являются:

1. Изучить работу метода Logit Lens на примере нескольких открытых мультимодальных моделей.
2. Выявить особенности формирования промежуточных логитов на разных слоях и проанализировать интерпретируемость этих состояний.
3. Сравнить результаты с уже опубликованными исследованиями в данной области и предложить улучшения для метода Logit Lens, повышающие качество интерпретации.

---

**Теоретическая часть**

Метод **Logit Lens** является одним из первых подходов, предложенных для интерпретации больших языковых моделей семейства GPT-2/3. Его ключевая идея — возможность «подсмотреть» распределение вероятностей (логиты), которое формируется внутри сети на промежуточных слоях.

В классической схеме генерации следующего токена в трансформерных моделях финальное распределение по словарю получается путём применения к последнему скрытому состоянию линейной проекции (обычно через блок Feed-Forward, состоящий из двух линейных слоёв), приводящей внутреннюю размерность к размеру словаря. Далее из этого распределения выбирается следующий токен (например, с помощью softmax и различных техник семплинга).

Logit Lens предлагает аналогичным образом смотреть на выход каждого промежуточного слоя, используя **ту же самую** обученную линейную «голову» (lm_head), которая обычно применяется лишь на финальном слое. Это позволяет в любой момент «развернуть» текущее скрытое состояние в пространство вероятностного распределения по словарю и увидеть, какие токены модель считает наиболее вероятными на данном этапе.

Таким образом, можно не только анализировать итоговые ответы модели, но и отслеживать её «мысли» на разных стадиях обработки входных данных. Например, при вопросе о столице России модель на ранних слоях может выдавать нерелевантные, но очень близкие по смыслу варианты вроде «Владимир», «Кремль» и т.п. Однако по мере продвижения к последним слоям распределение сдвигается к правильному ответу «Москва» (с учётом особенностей разбивки на токены).

**Преимущества метода Logit Lens**:

- Прозрачность процесса генерации. Можно наглядно увидеть, как модель формирует гипотезы.
- Простота реализации. Не требуется дополнительное обучение; используется уже обученная линейная голова.
- Универсальность. Метод можно адаптировать к различным архитектурам и сценариям применения.

Однако у метода есть ограничение: поскольку линейная голова обучалась под последние скрытые состояния, ранние слои могут содержать «сырой» сигнал, не предназначенный напрямую для финального декодирования. Из-за этого топ-токены с ранних слоёв зачастую выглядят неинтерпретируемыми.

---

**Методология**
В ходе исследования рассматривались следующие модели:

1. **llava-hf-7B**
2. **llama-3.2-11B-vision**

Все эксперименты проводились на A40 48gb.

В качестве датасета для экспериментов взяты 5 изображений из набора COCO-val2017. Картинки подбирались так, чтобы они отличались формами, цветами и присутствующими на них объектами.

Для интеграции Logit Lens в каждую модель потребовалось учесть особенности их работы с визуальными токенами:

- **Llava** использует механизм, при котором в последовательность добавляются специальные токены `<image_pad_id>`, число которых зависит от того, как изображение делится на патчи. Эти токены позднее заменяются на скрытые состояния из модуля `vision_encoder`.
- **llama-3.2-11B-vision** использует иной подход: для изображения у неё предусмотрен единственный токен, а связи с визуальным модулем обеспечиваются через механизм `cross_attention_states`, так как архитектурно Llama разделяет текстовую и визуальную часть, используя между ними кросс-аттеншен.

После учёта нюансов мультимодальной части во всех моделях к скрытым состояниям промежуточных слоёв последовательно применялся уже обученный слой `lm_head` (при необходимости вместе с `norm`). Это и позволило получить распределения логитов на каждом слое и оценить, какие токены модель «предпочитает» в каждый момент времени.

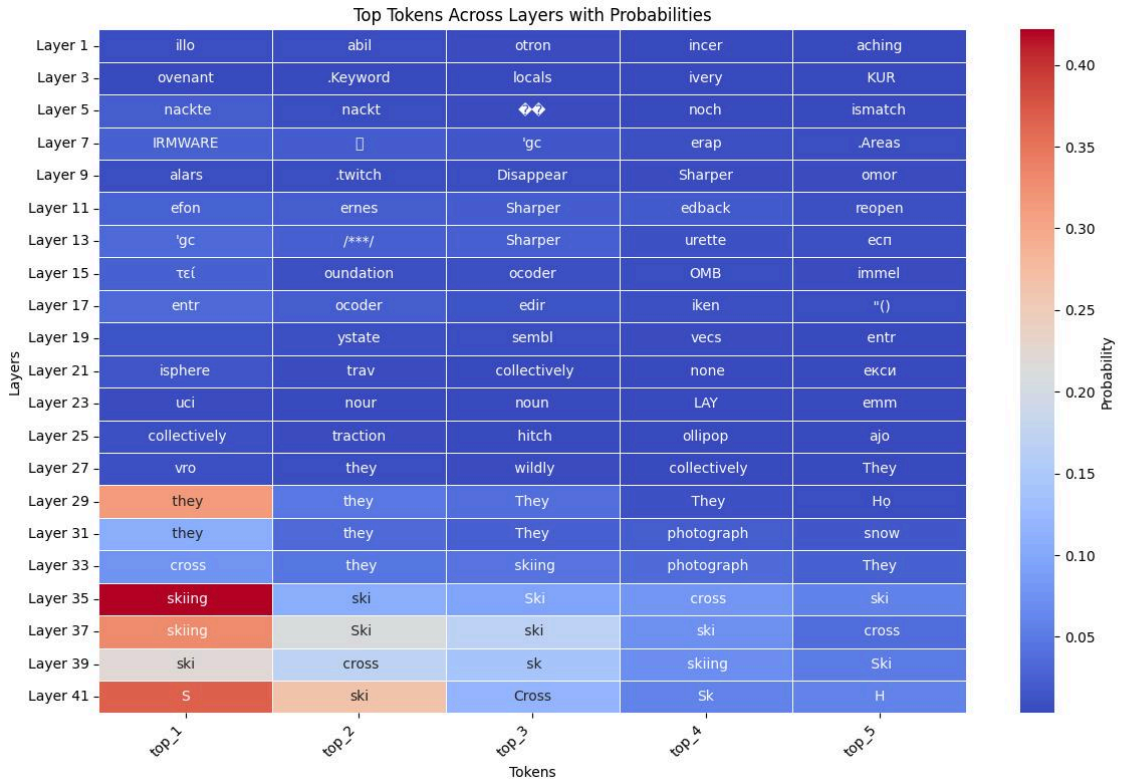
---

**Экспериментальная часть**

Для начала экспериментов был проведён стандартный инференс на модели Llama. Цель — проследить, как меняется топ-k вероятных токенов в зависимости от глубины слоя. Рассматривался последний токен последовательности, для которого и предсказывались вероятности следующих токенов.

(рисунок 1)

В качестве примера брался следующий промпт к изображению (рис. 1):
*“What are they doing? Answer with one word.”*




(график 1)

На выходе получено распределение токенов по слоям (график 1). Как видно из графика, разумные словоформы начинают появляться ближе к 25-му слою, а ожидаемый по смыслу токен (или его вариации) — примерно на 33-м. При этом на 31-м слое в топ-5 входит слово «снег», что может свидетельствовать о том, что модель ещё не полностью «сфокусировалась» на правильном контексте.

Далее был воспроизведён эксперимент, аналогичный описанному в [1] (см. arxiv:2410.07149v1). В `input_ids` дополнительно добавляются токены для каждого патча, что упрощает анализ: мы можем проследить, «о чём думает» модель для каждого кусочка изображения.

Код эксперимента практически не отличается от предыдущего, за исключением того, что теперь рассматривается предсказание токена для **каждого** элемента входной последовательности, включая визуальные патчи. Для удобства визуализации использовался скрипт авторов статьи, позволяющий сопоставлять фрагмент изображения и предсказываемые моделью токены в HTML-формате (пример: `vis/llava-1.5-7b-hf_3_logit_lens.html`).

Результаты показывают, что на самых ранних слоях модель генерирует малоинформативные или не связанные с изображением токены. На средних слоях (около 14–16) уже появляются более осмысленные слова: для жирафа встречаются «animal», «pattern», «hair», «neck». В окрестности патча с травой могут появляться токены «background», «field», «green» и т.д.

В ходе эксперимента также отмечено, что некоторые ранние паттерны исчезают и заменяются другими. Так, части жирафа, относящиеся к шее, сначала были ассоциированы с токенами «animal» и «pattern», но позднее модель начала чаще использовать «neck». Интересный момент — появление немецких токенов, таких как «Augen» («глаза») на 16-м слое, в то время как английский вариант «eyes» появляется лишь ближе к 22-му слою.

---

**Анализ проблем и предложенные решения**

1. **Проблема неинтерпретируемых токенов на ранних слоях**
   Все протестированные модели в ранних слоях выдают набор слов, который либо кажется «мусором», либо содержит лексемы на разных языках, никак не связанных с изображением (или будущим контекстом). Это объясняется тем, что линейная голова (`lm_head`) обучена под последние скрытые состояния, а не под промежуточные.

   **Предложенное решение**:

   - Ввести небольшой обучаемый адаптер для каждого (или нескольких) скрытых слоев, чтобы приводить выходные вектора к пространству, более близкому к финальным скрытым состояниям.
   - Обучать этот адаптер на задаче предсказания следующего токена с помощью кросс-энтропии (или минимизации KL-дивергенции между распределением на текущем слое и итоговым распределением).
2. **Плюсы**:
   - Увеличение интерпретируемости, так как адаптер «подгоняет» ранние слои под пространство финального предсказания.
   - Если адаптер может восстановить финальное распределение, значит, ранние слои уже содержат значимую информацию о будущем ответе.
3. **Минусы**:
   - Возможна «переподгонка» к итоговому распределению, искажающая реальное содержимое скрытых состояний.
   - Увеличение вычислительной нагрузки из-за обучения адаптеров для каждого слоя.
4. **Метод TunedLens**
   В работе [2] (arxiv:2303.08112) авторы развивают идеи Logit Lens и предлагают так называемый TunedLens — аффинное преобразование скрытых состояний на каждом слое (обучаемая матрица и bias), помогающее «сместить» пространственные различия между ранними и поздними слоями. Они мотивируют введение обучаемого смещения (bias) тем, что остаточные соединения (residual connections) могут существенно сдвигать вектора по мере прохождения слоя, и компенсировать этот сдвиг нужно явно.

---

**Заключение**

В ходе исследования было продемонстрировано, как метод Logit Lens может быть интегрирован в различные мультимодальные LLM (Llava-hf-7B, llama-3.2-11B-vision). Анализ полученных логитов на промежуточных слоях позволил сделать вывод о том, что:

1. На ранних слоях модели генерируют шумовые и малозначимые токены, связанные с тем, что линейная голова не обучалась на этих представлениях.
2. На средних и последних слоях появляются более интерпретируемые токены, отражающие контекст и фактическое содержание изображения.
3. Добавление обучаемых адаптеров или использование подхода TunedLens способно повысить интерпретируемость промежуточных логитов и приблизить их к финальному распределению, сохранив при этом причинную достоверность (causal fidelity) интерпретации.

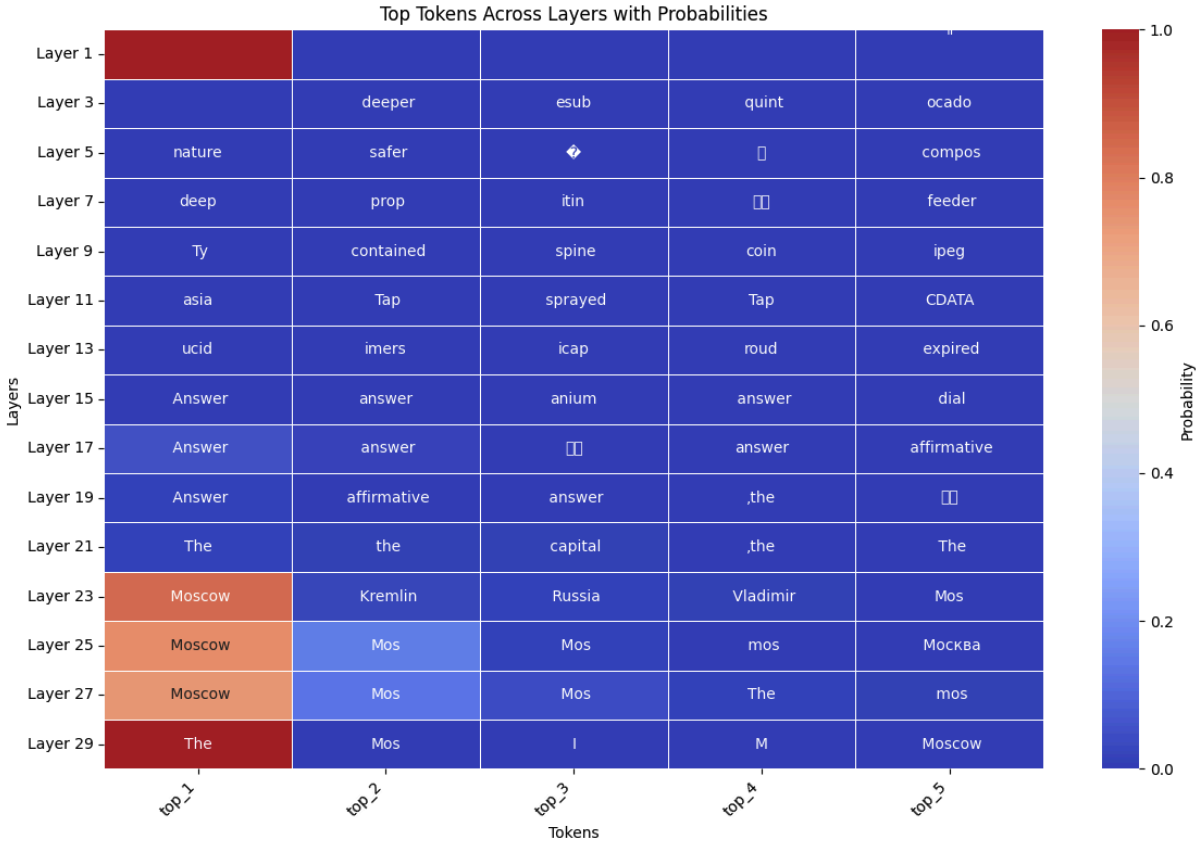
---

**Приложения**

1. **Код реализации**

   Исходный код с примерами интеграции Logit Lens в различные модели, а также скрипты для визуализации можно найти в репозитории:
   github.com/ponik7/multimodal-test-task

2. **Дополнительные графики и таблицы**




(График 2. Промпт: “Какой город является столицей России”)

[1] Neo, C., Ong, L., Torr, P., Geva, M., Krueger, D., & Barez, F. (2025). **Towards Interpreting Visual Information Processing in Vision-Language Models**. *Nanyang Technological University*, *University of Oxford*, *Tel Aviv University*, *MILA*, *Tangentic*. Work conducted during ERA-Krueger AI Safety Lab internship. // arXiv:2410.07149v1

[2] Belrose, N., Ostrovsky, I., McKinney, L., Furman, Z., Smith, L., Halawi, D., Biderman, S., & Steinhardt, J. (2025). **Eliciting Latent Predictions from Transformers with the Tuned Lens**. // arXiv:2303.08112