

# In-context learning and Induction Heads

Докладчик: Лишуди Дмитрий



# Мотивация

- Современные языковые модели умеют имитировать умные алгоритмы **in-context**.
- Но нейросети - серый ящик.
- Для их изучения стоит использовать реверс-инжиниринг.
- Утверждаются, что эти алгоритмы реализуются **индукционными головами** Attention'a.



# In-context learning

Few-shot?

1	Translate English to French:	← task description
2	sea otter => loutre de mer	← examples
3	peppermint => menthe poivrée	←
4	plush girafe => girafe peluche	←
5	cheese => .....	← prompt

# In-context learning

Few-shot?

Больше контекста - проще предсказать.



$$L(t_{500}, \underbrace{model(text[: 500])}_{\hat{t}_{500}}) - L(t_{50}, \underbrace{model(text[: 50])}_{\hat{t}_{50}})$$



## Индукционные головы

Это такие головы трансформера, что для произвольных (случайных) последовательностей токенов имеет свойства:

- **Нахождение префикса:** голова *обращает внимание* на токены, которые следовали за текущими.
- **Копирование:** голова дает больший вес токенам, на которые обращено внимание.

То есть реализует базовый индукционный вывод:

$[A][B] \dots [A] \rightarrow [A][B] \dots [A] [B]$



# Устройство трансформера

- **Decoder-only:** подаем последовательность токенов, предсказываем следующий.
- На практике чередуются Multi-Head Attention и MLP, мы в основном исследуем **без MLP**.
- **Residual:** к исходному представлению  $X$  последовательно добавляем выходы слоев.
- В конце идет линейный слой и softmax для логитов следующего токена.

## One Head Attention

$$W_Q, W_K, W_V, W_O \in \mathbb{R}^{d \times n}; \quad X \in \mathbb{R}^{n \times L}; \quad H(X) \in \mathbb{R}^{n \times L}$$

$$K = W_K X \quad A = \text{softmax}(Q^T K) = \text{softmax}\left(X^T \underbrace{W_Q^T W_K}_{W_{QK}} X\right) \in \mathbb{R}^{L \times L}$$

$$Q = W_Q X$$

Матрица  
внимания

$$V = W_V K$$

$$H(X) = W_O^T V A = \underbrace{W_O^T W_V}_{W_{OV}} X A$$

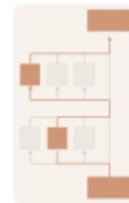
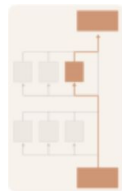
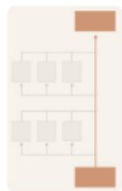
Токен - Токен

Позиция - Позиция

## Комбинация Attention

$$T_i(X) = X + \sum_{h \in H_i} H(X)$$

$$T_2(T_1(X)) = X + \sum_{h \in H_1 \cup H_2} h(x) + \sum_{h_1 \in H_1} \sum_{h_2 \in H_2} W_{OV}^{h_2} W_{OV}^{h_1} X A_{h_1} A_{h_2}$$

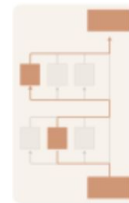
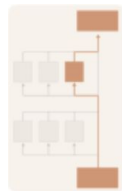
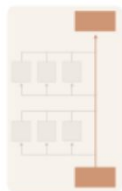




## Комбинация Attention

$$T_i(X) = X + \sum_{h \in H_i} H(X)$$

$$T_2(T_1(X)) = X + \sum_{h \in H_1 \cup H_2} h(x) + \sum_{h_1 \in H_1} \sum_{h_2 \in H_2} W_{OV}^{h_2} W_{OV}^{h_1} X A_{h_1} A_{h_2}$$



Матрица A зависит от T\_1(X)!



## Композиции в А

$$\begin{aligned} A_{h_2} &= \text{softmax}\left((X + h_1(X))^T W_{QK} (X + h_1(X))\right) \\ &= \text{softmax}\left(X^T W_{QK} X + A_{h_1}^T X^T (W_{OV}^{h_1})^T W_{QK} W_{OV}^{h_1} X A_{h_1} \right. \\ &\quad \left. + A_{h_1}^T X^T (W_{OV}^{h_1})^T W_{QK} X + X^T W_{QK} W_{OV}^{h_1} X A_{h_1}\right) \end{aligned}$$

# Откуда берутся индукционные головы

out about the Potters. Mrs Potter was ... neighbours would say if the Potters arrived in

out about the Potters. Mrs Potter was ... neighbours would say if the Potters arrived in

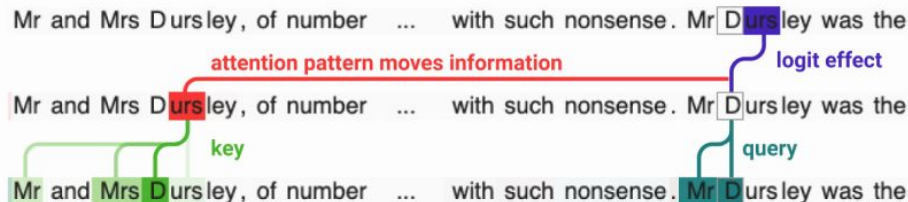
out about the Potters. Mrs Potter was ... neighbours would say if the Potters arrived in



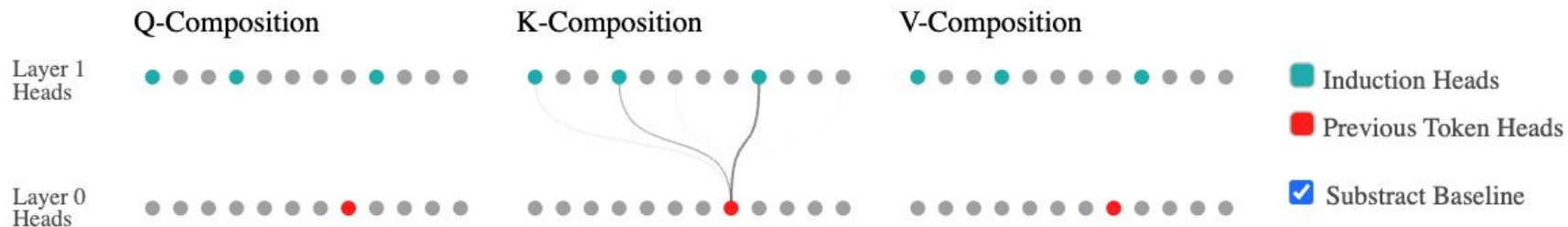
Mr and Mrs Dursley, of number ... with such nonsense. Mr Dursley was the

Mr and Mrs Dursley, of number ... with such nonsense. Mr Dursley was the

Mr and Mrs Dursley, of number ... with such nonsense. Mr Dursley was the



# Влияние композиций



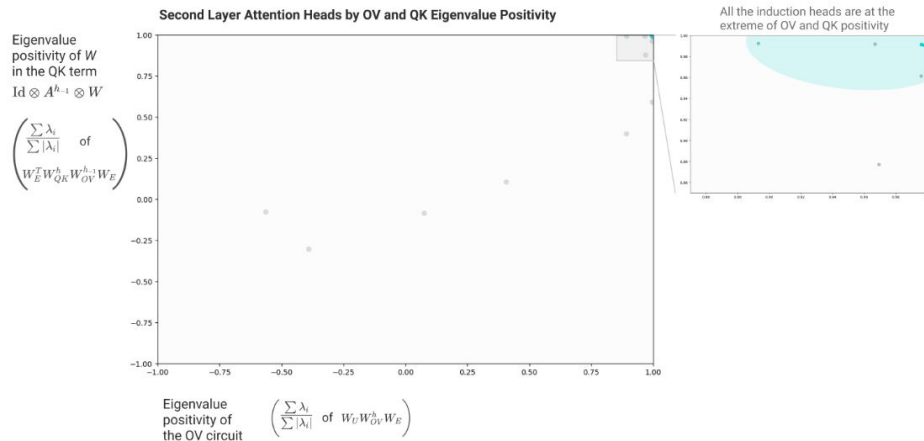
Сила связи зависит от коэффициентов вида:


$$\frac{\|W_{QK}^{h_1}W_{OV}^{h_1}\|_F}{\|W_{QK}^{h_1}\|_F\|W_{OV}^{h_1}\|_F}$$

# Реверс-инжиниринг маленьких моделей явно показывает на связь с in-context.

Аргумент №1

- Сам факт существования определенных нами индукционных голов подразумевает хорошее in-context обучение.
- При этом мы действительно знаем их внутреннее устройство для малых моделей.





# Индукционные головы способны на сложные алгоритмы

Аргумент №2

- Есть головы, имплементирующие сложные индукционные алгоритмы:

$[A^*][B^*] \dots [A] \rightarrow [A^*][B^*] \dots [A] [B]$

- $A^*$  похожа на  $A$ ,  $B^*$  похожа на  $B$ .
- При этом они подходят под критерий индукционных голов.
- Большие модели создают абстрактные представления, которые позволяют получить такие алгоритмы



## Метрики “индукционности” головы

- **Нахождение префикса:** Генерируем случайную последовательность, повторяем 4 раза. Вычисляем  $A$ , усредняем веса соответствующим правильному продолжению.
- **Копирование:** Генерируем случайную последовательность, смотрим вывод головы, преобразуем последним слоем модели в логиты. Из логитов вычитаем среднее и прогоняем через ReLU. Берём отношение искомого логита ко всем остальным, нормируем.
- **Предыдущий токен:** Выбираем случайный объект из тренировочной выборки. Считаем  $A$ , усредняем веса соответствующие парам токен  $i \leftrightarrow$  токен  $i-1$ .



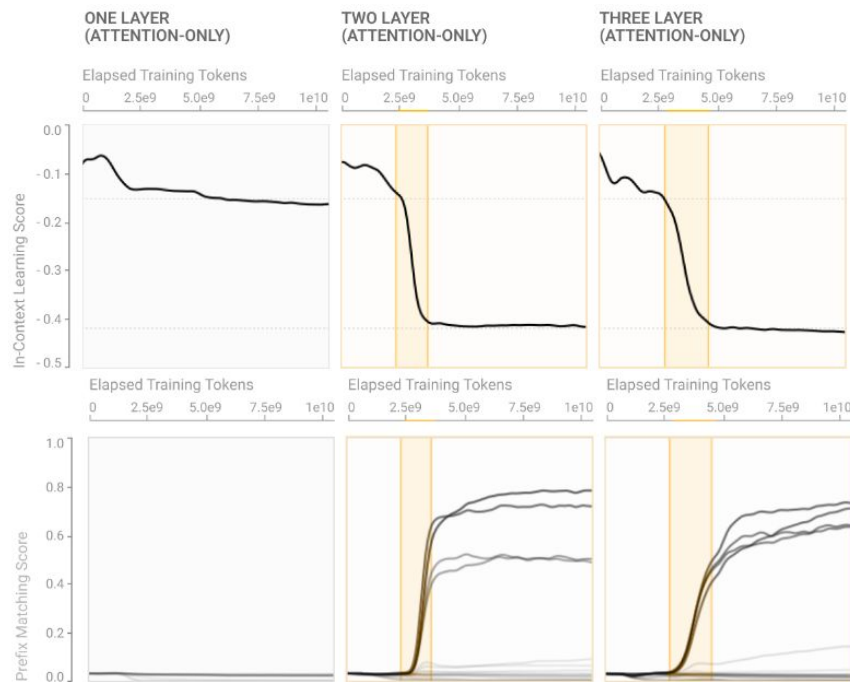
# Одновременные фазовые переходы

Аргумент №3

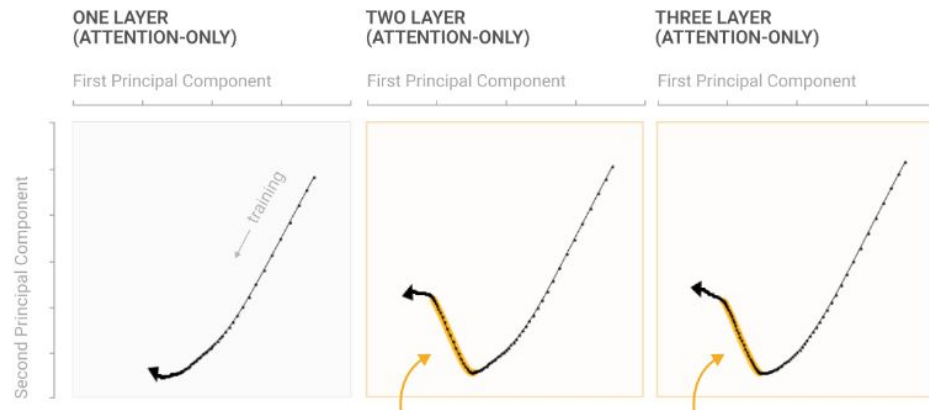
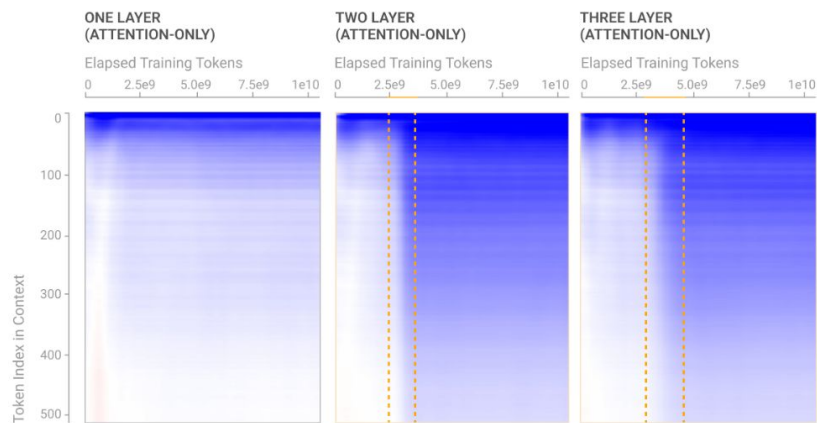
- Во время обучения трансформеры проходят фазовый переход.
- В этот фазовый переход метрика in-context значительно падает.
- В то же время у модели появляется большое число индукционных голов.



# Фазовый переход



# Фазовый переход



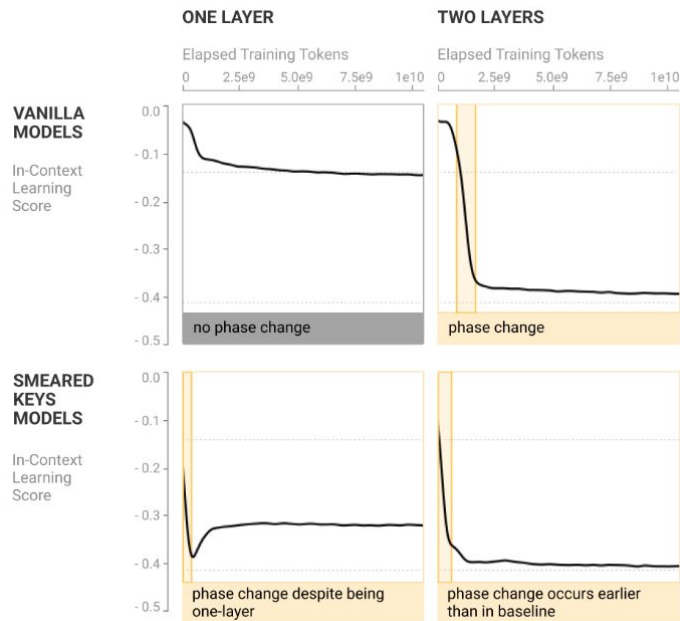
# Чем легче архитектуре создать индукционные головы, тем лучше in-context

Аргумент №4

- Добавляем “размазывание” ключей:

$$k_j^h = \sigma(\alpha^h)k_j^h + (1 - \sigma(\alpha^h))k_{j-1}^h$$

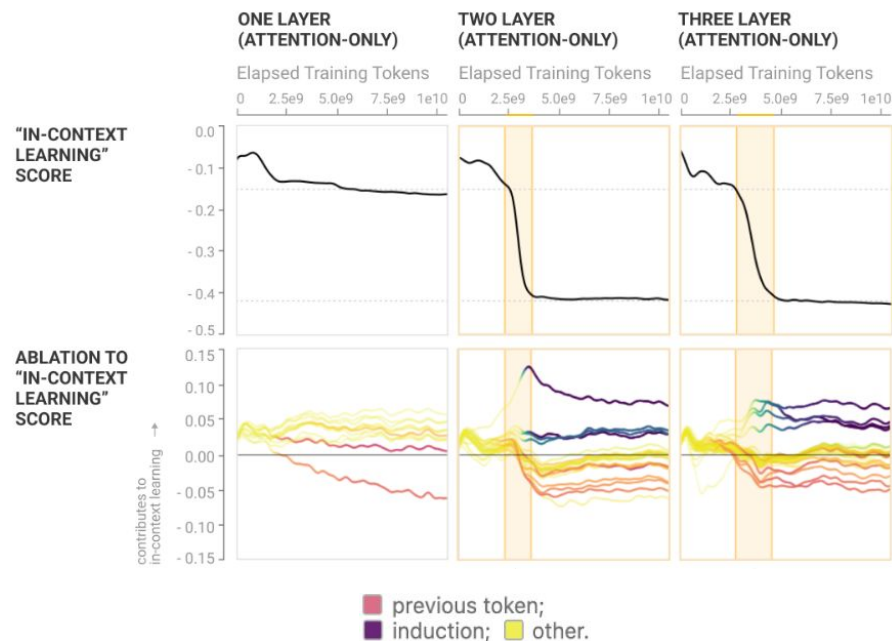
- Теперь даже однослойные модели могут создать индукционные головы.




# In-context обучение сильно ухудшается при удалении индукционных голов.

Аргумент №5

Убираем случайные головы, смотрим  
как влияет на метрики в зависимости  
от головы.





## **Предыдущие аргументы можно экстраполировать на сложные модели.**

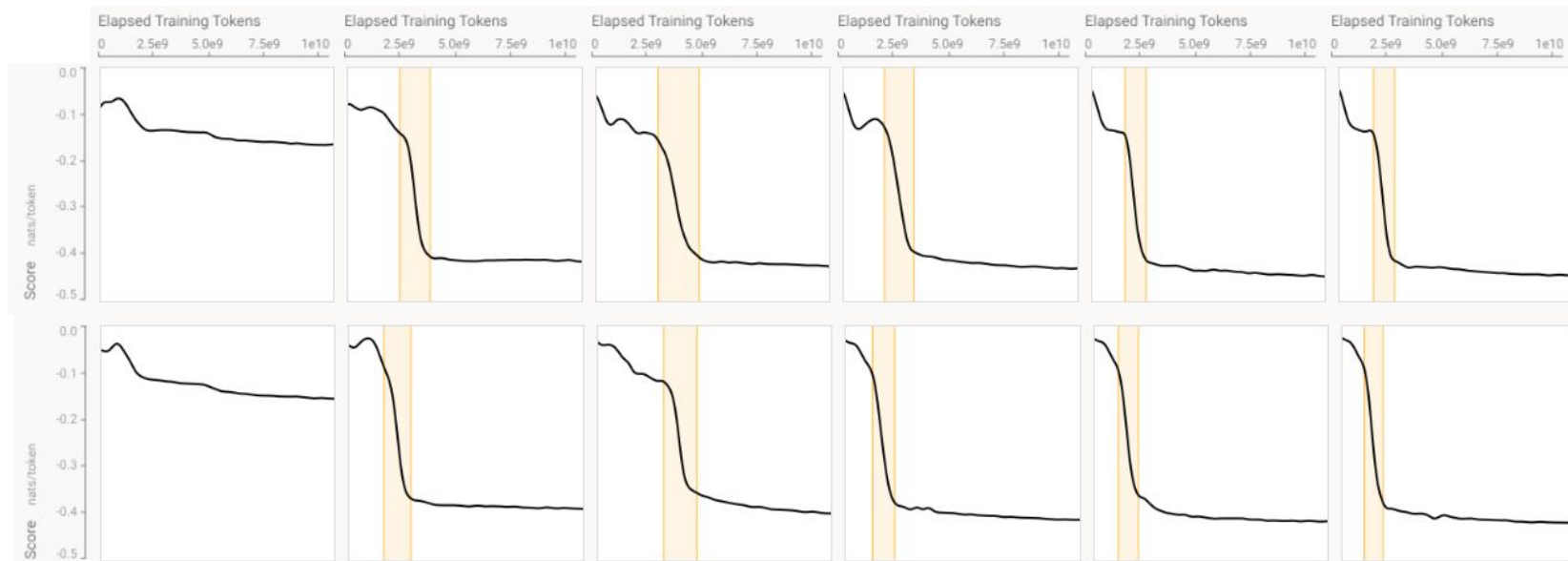
Аргумент №6

- В прошлых аргументах исследовались двухслойные трансформеры без MLP.
- Явно обобщить те аргументы на сложные модели нельзя.
- Можно показать, что по многим показателям сложные модели ведут себя так же, как простые.
- Графики из фазовых переходов получаются почти одинаковыми.

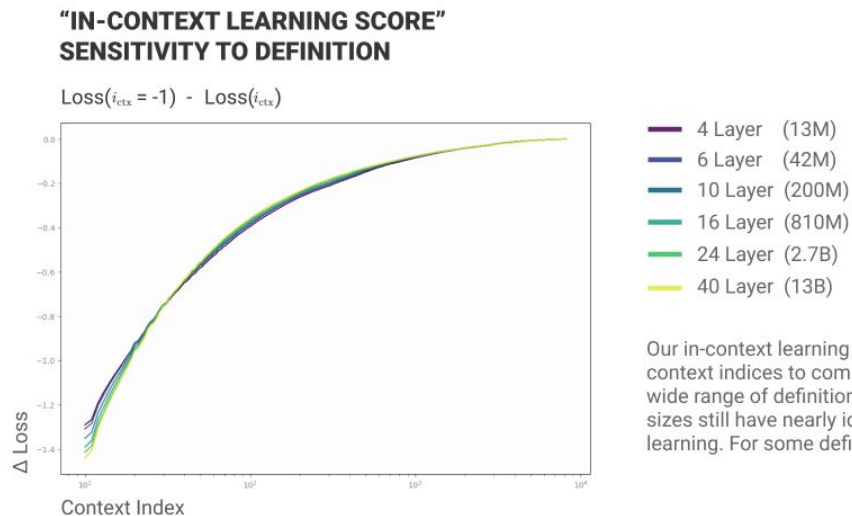
# Экстраполяция на сложные модели

Нет  
MLP

Есть  
MLP



# Влияние размера на метрику in-context



Our in-context learning score requires one to pick two context indices to compare. But it appears that for a wide range of definition choices, models of different sizes still have nearly identical amounts of in-context learning. For some definitions, small models do more.



## Заключение

- Особенные комбинации голов явно реализуют умные алгоритмы.
- Большая часть аргументов прямо относится лишь к маленьким моделям без MLP слоев.
- Глубокие трансформеры способны значительно усложнять эти алгоритмы.
- Фазовый переход появления индукционных голов может быть связан со многими DL эффектами.