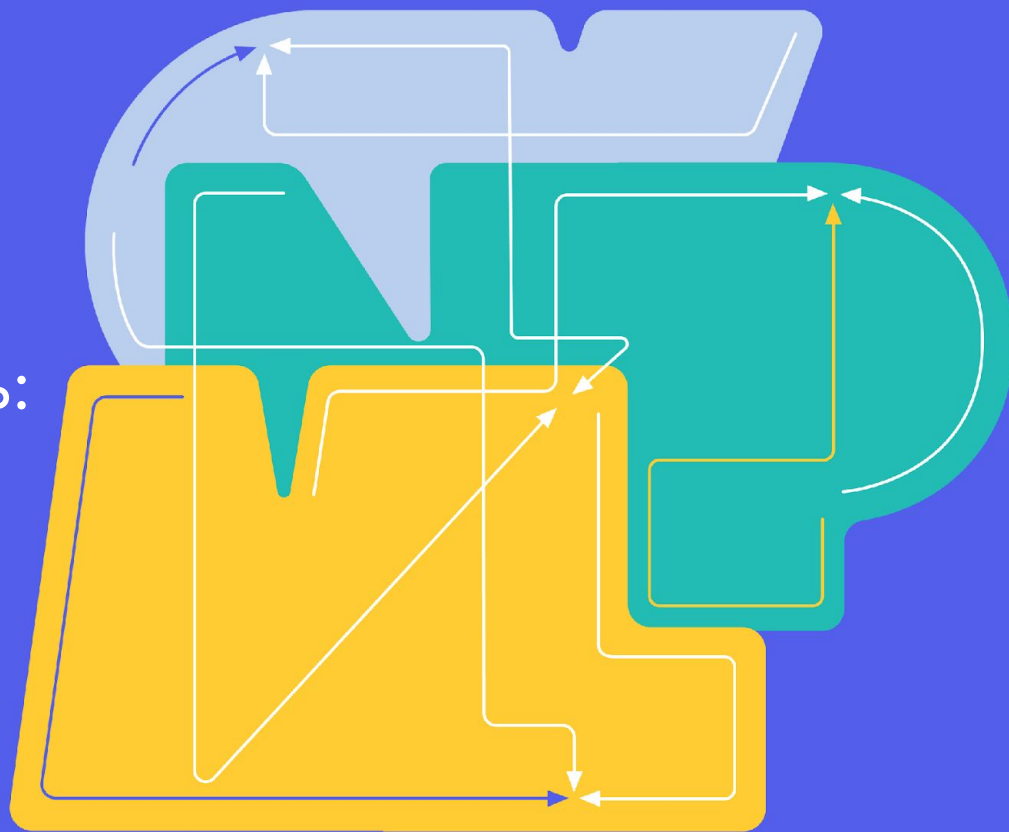




Научно-образовательный  
центр  
Томский  
государственный  
университет

# Мета-обучаемая память: Test-Time Training

Матвей Каиров, Даниил Сергеев



# Содержание

---

01 Мета-обучение и  
In-Context Learning

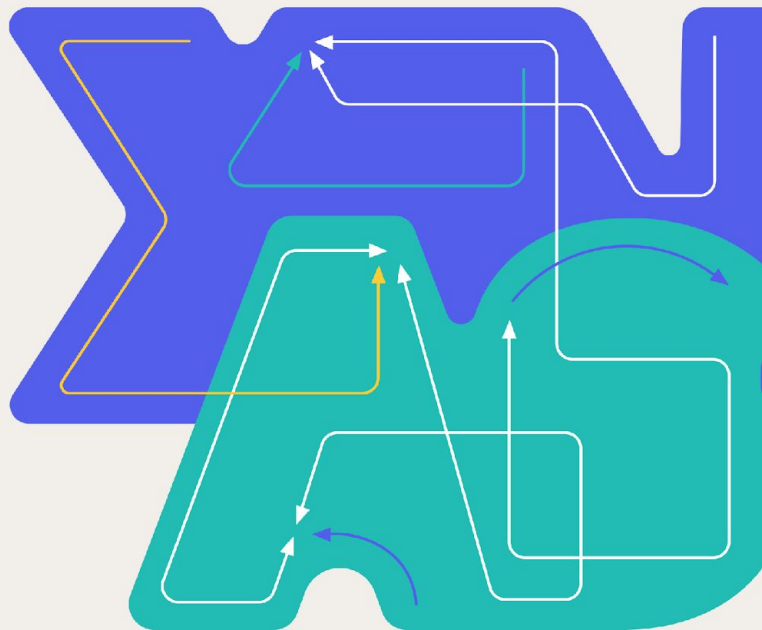
02 Методология

03 Результаты

# 01

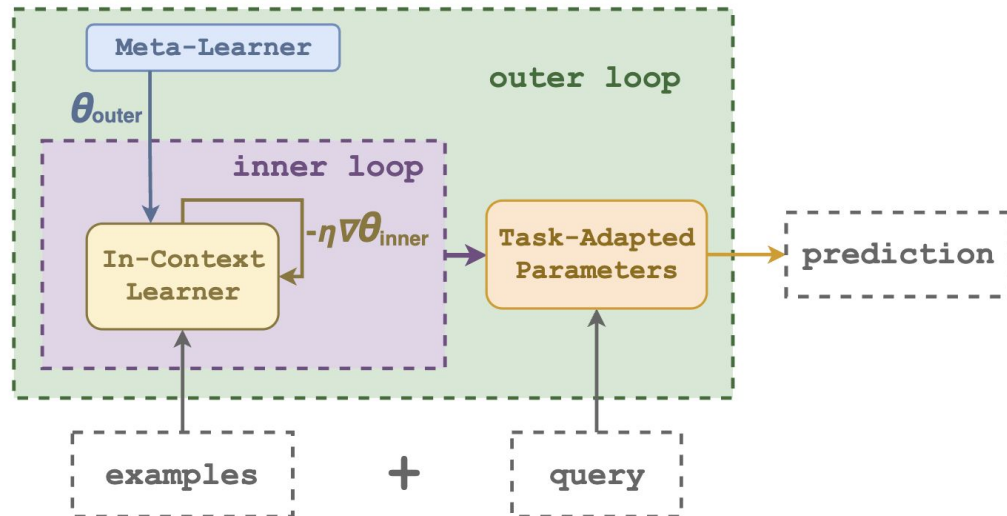
---

## Мета-обучение и In-Context Learning



# Общая концепция

- Вместо In-Context Learning проводим явные обновления параметров градиентным спуском.
- Модель проводит несколько итераций адаптации к полученным данным
- Внешний мета-учитель оптимизирует гиперпараметры внутреннего цикла, инициализации и вспомогательные параметры, чтобы пользоваться мета-памятью.



Model-Agnostic Meta-Learning. Градиентный спуск для внешних и внутренних  $\theta$  проводится отдельно.

# Приложения в QA-задачах

→ Можно использовать во few-shot question-answering задачах

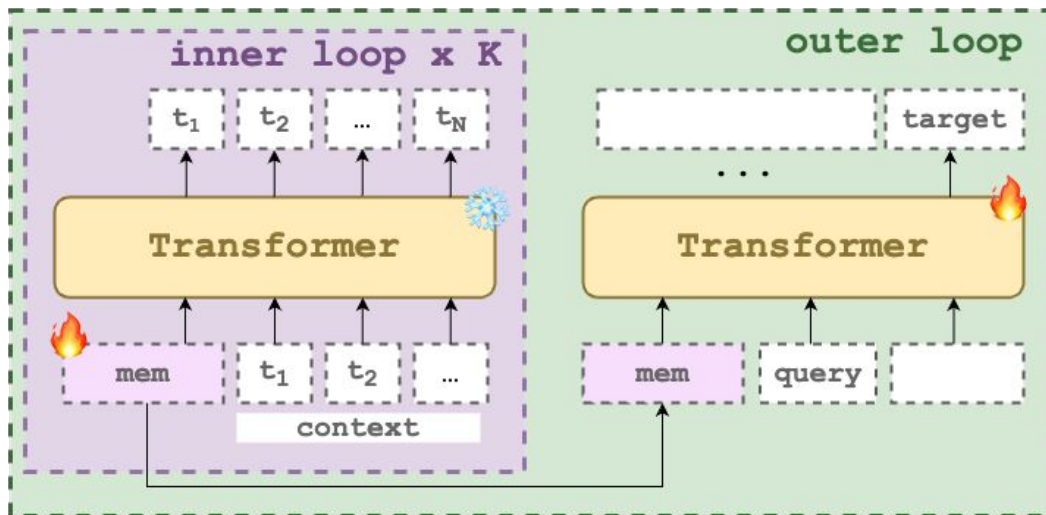
→ Адаптируем параметры во внутреннем цикле к примерам

→ После адаптации можем давать ответы на вопросы

→ Быстрый retrieval информации из памяти

→ Не нужно заново подавать модели инструкции и контекст

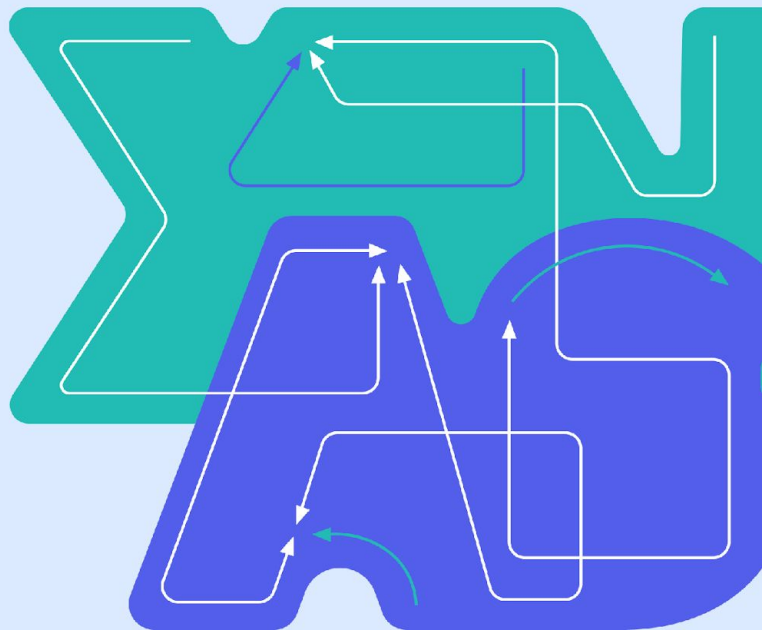
→ Доставать информацию из памяти эффективнее чем KV Caching.



Test-Time Training в QA-задачах. С помощью prompt-tuning обучаем память на LM-задачу во внутреннем цикле, после чего с ее помощью отвечаем на вопрос по контексту.

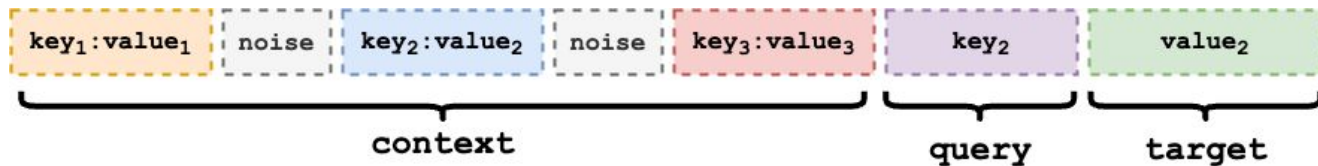
# 02

## Методология



# Данные для обучения

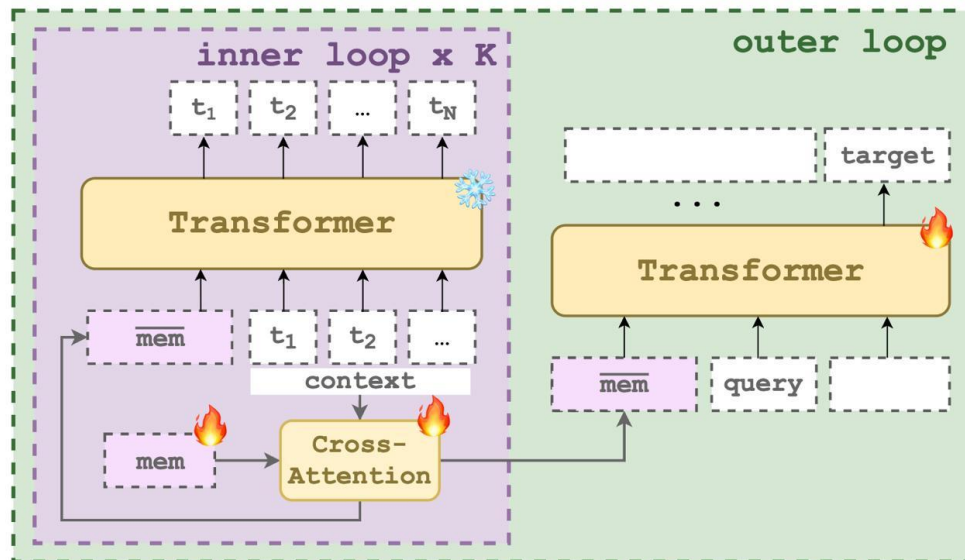
- Используем задачу Associative Retrieval.
- Случайно генерируются пары ключ-значение.
- Между парами вставляется шум.
- Задача модели – по данному ключу вспомнить ассоциированное с ним значение.
- Последовательность с парами и шумом запоминается во внутреннем цикле, во внешнем цикле подается запрос с ключом.



Associative Retrieval со встроенным шумом. Контекст содержит много пар, разделенных шумом, запрос и цель – одна из встретившихся в контексте пар.

# NeuralMemory

- Обучаем memory токены и параметры NeuralMemory в inner loop
- Обучаем LLM и  $\overline{mem}$  в outer loop
- Память записывается в memory токены через cross-attention
- $q$  - проекция memory токенов
- $k, v$  - проекции эмбединга контекста

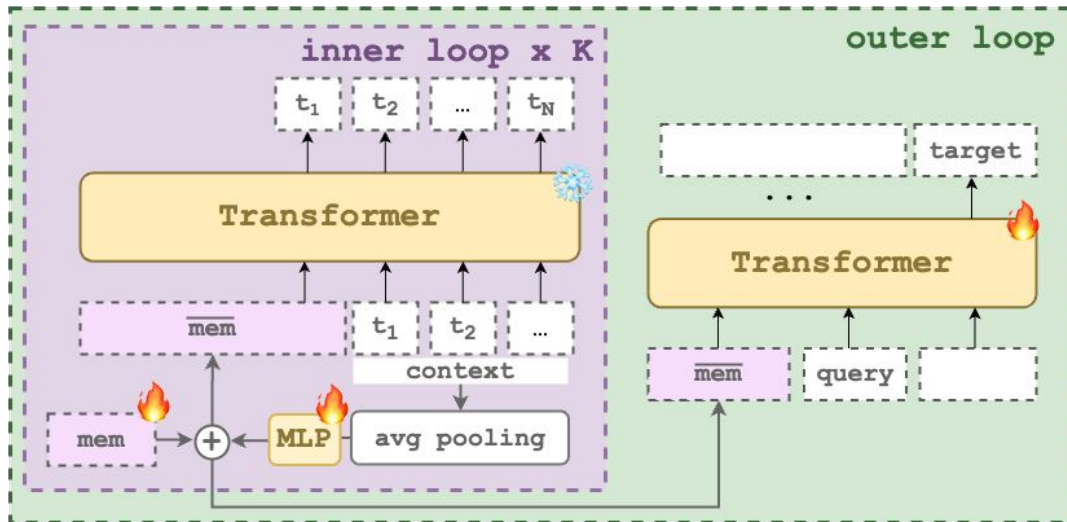


Meta-обучаемая память. Внутренние параметры учатся восстанавливать последовательность через замороженный backbone, внешние отвечают на запросы по сжатому контексту.



# MemoryMLP

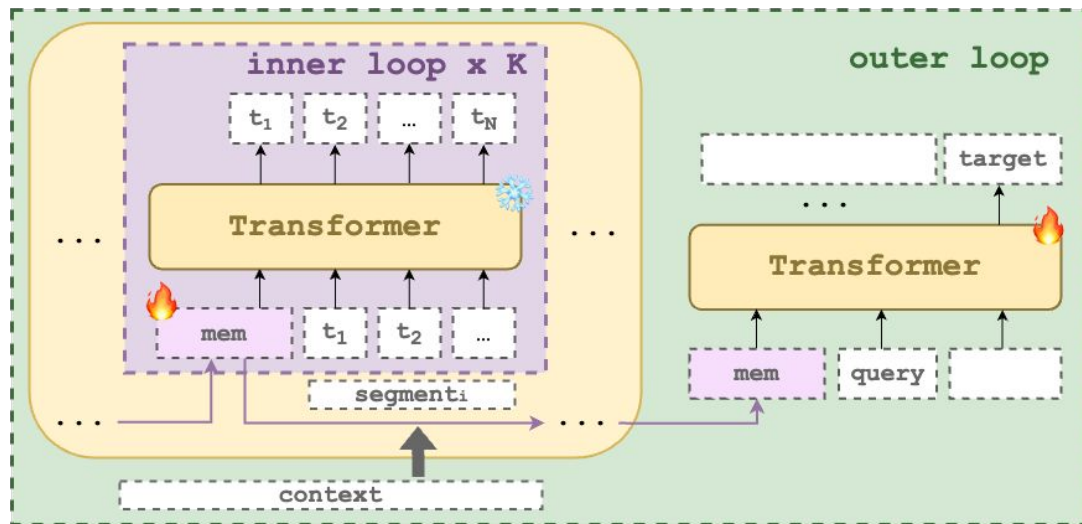
- Обучаем memory токены и параметры MemoryMLP в inner loop
- Обучаем LLM и  $\overline{\text{mem}}$  в outer loop
- Подаем context embedding в MemoryMLP
- $\text{memory} = \text{mem} + \text{MLP}(\text{ctx\_emb})$



Meta-обучаемая память. Внутренние параметры учатся восстанавливать последовательность через замороженный backbone, внешние отвечают на запросы по сжатому контексту.

# MAML + RMT

- Можно не запоминать всю последовательность разом, а разбить ее на сегменты.
- Токены памяти накапливают изменения через градиентный спуск по сегментам.
- На финальном сегменте убираем цикл адаптации и учимся доставать информацию из памяти.



RMT with Test-Time Training (RMT4). Мета-обучение проводится на каждом сегменте последовательности, изменения градиентным спуском накапливаются в токенах памяти.

# 03

---

## Результаты



## Вариации Meta-памяти

Датасет \ Модель	Neural Memory	Memory MLP
N1-K4V4-S1_16-32_1M	0.999	0.947
N2-K4V4-S2_16-32_1M	0.721	0.045

Лучшая точность предсказания пар на датасетах.

# RMT с Meta-памятью

Датасет \ Модель	RMT4	Parallel RMT4
N1-K4V4-S1_16-32_1M	0.99	0.06
N2-K4V4-S2_16-32_1M	0.0	0.0

Лучшая точность предсказания пар на датасетах.

# Future Work

## → Обучение мета-памяти на сложные задачи

- Подобрать гиперпараметры для моделей.
- Обучить модели на BABILong и LM-задачи.
- Добавить LoRA адаптер в качестве обучаемой памяти.
- Попробовать обучить модели с помощью расписания.

## → Сжатие контекста RMT мета-памятью

- Поработать над скоростью сходимости
- Обучить послойную память для интеграции модели с Diagonal Batching



[airi.net](https://airi.net)



[airi\\_research\\_institute](https://t.me/airi_research_institute)



[mkairov](https://t.me/mkairov)



[AIRI Institute](https://vk.com/AIRI_Institute)



[danilka200300](https://t.me/danilka200300)

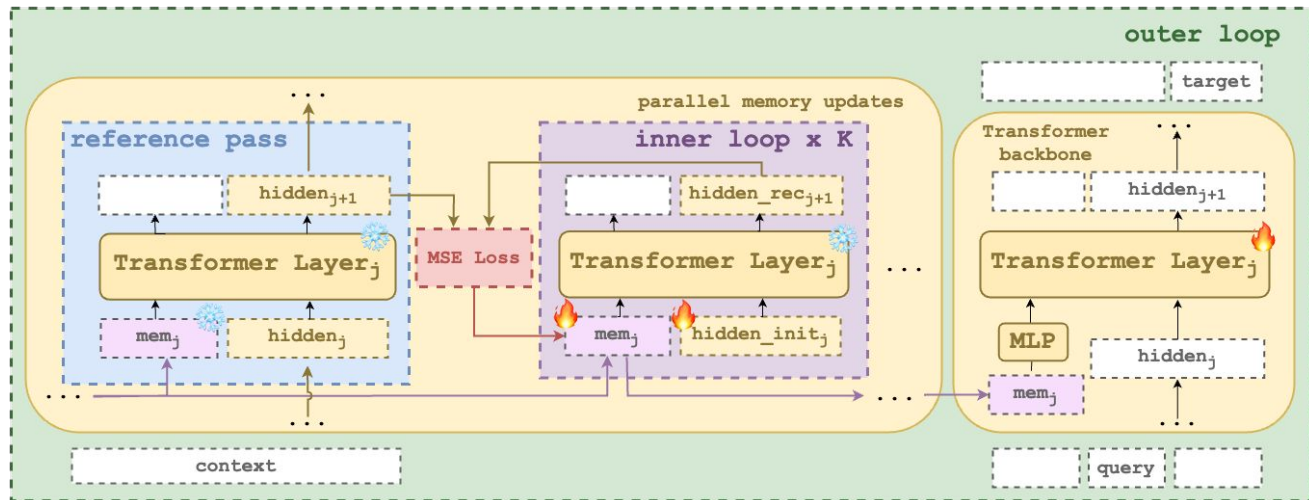


Telegram

AIRI

# MAML + RMT + Parallel memory updates

- Вспомним послойную память в ARMT
- Будем обучать память восстанавливать hidden'ы на каждом слое
- Подменяем таргет ТТТ-обучения: на каждом слое вместо LM-лосса по контекстной и глобальной памяти восстанавливаем hidden и считаем MSE с настоящим выходом слоя.



Parallel RMT with Test-Time Training (PRMT4). Мета-обучение проводится на каждом сегменте последовательности и каждом слое трансформера. Хорошо поддается параллелизации (не нужно дожидаться окончания шагов в inner loop для подсчетов на следующем сегменте + Diagonal Batching).