

От слабого ИИ к общему универсальному интеллекту (обзор тенденций 2020-2023)

Визильтер Юрий Валентинович, д.ф.-м.н., проф. РАН, директор по направлению – руководитель научного комплекса «Искусственный интеллект и техническое зрение» ФАУ «ГосНИИАС», viz@gosniias.ru

Научно-исследовательский семинар
РАИИ и ФИЦ ИУ РАН
«Проблемы искусственного интеллекта»
Москва, ФИЦ ИУ РАН, 31.01.2024



Доклад на МКПУ-2019: состояние нейросетевых технологий ИИ и их применение в задачах обработки информации и управления (2019)

Использование глубоких нейронных сетей для анализа данных, управления и оптимизации в перспективных авиационных приложениях

Визильтер Юрий Валентинович, д.ф.-м.н., проф. РАН, viz@gosniias.ru
Желтов Сергей Юрьевич, д.т.н., академик РАН, zhl@gosniias.ru

ФГУП «ГосНИИАС»

12-я Мультиконференция по проблемам управления (МКПУ-2019)



с. Дивноморское, Геленджик, Краснодарский край
24 сентября 2019 г

Как развивались технологии «ИИ», чего достигли, чего ждать?

Ранний период развития ИИ: созданы основные методы и подходы...



...2011+: в области ИИ началась технологическая революция ГНС

Алгоритмическое обеспечение, необходимое для автономных мобильных систем (РТК, БЛА,...)



В 2019 г. мы уже рассказывали о состоянии технологий ИИ...

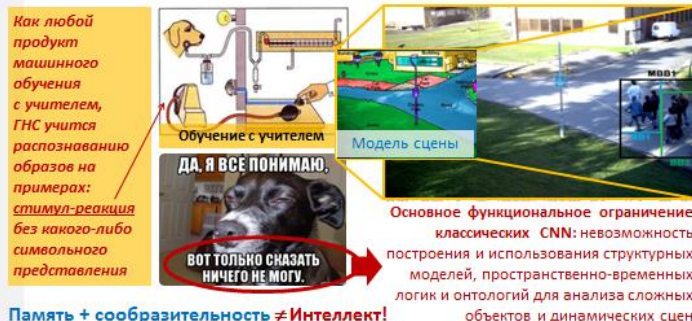
...о возникновении ГНС и 2 волнах технологической революции ИИ...

...о том, что ряд задач создания ИИ для РТК уже решен CNN ...

- Успехи и ограничения CNN
- Необходимость объединения нейросетей со структурными моделями, логикой и БЗ
- Роль обучения с подкреплением в создании функционального ИИ для задач автономного и группового управления

Почему будущее не наступило уже в 2015 году?

Почему распознавания образов не хватило для создания ИИ?
(чего не удалось добиться в рамках первой волны технологической революции)



Events detection using a video-surveillance Ontology and a rule-based approach,
Yassine Kazi Tani, Adel Lablack, Abdelghani Ghomari, and Ioan Marius Bliascu, 2014

2019: Прогноз создания функционального ИИ (для автономных мобильных систем – РТК, БЛА,...)



...что CNN ограничены без работы со структурами, логикой и БЗ...

...что DL + RL обеспечат с 2020 путь к функциональному ИИ для РТК

Тема доклада: как развитие технологий ИИ (2020-23) влияет на решение задач робототехники и управления

• Основные тенденции развития ИИ (2020-23)

- *Computer Vision: CNN → Transformers*
- *NLP: GPT-3, Foundation Models, GPT-4*
- *RL: Open-Ended Learning, GATO, GITM, Verbal RL*
- *Learning → Prompt Engineering (+Learning)*
- *Универсальные модели: VLA, UniAD, Socratic Models*

• Применение ИИ в авиационных комплексах

- *создание отечественной платформы внедрения ИИ*
- *интеллектуализация ВС ГА*

Где мы
наблюдаем
или
ожидаем
смену
парадигмы

Обработка
информации

Управление

ИИ2 → ИИ1+ИИ2

Обработка-
Управление

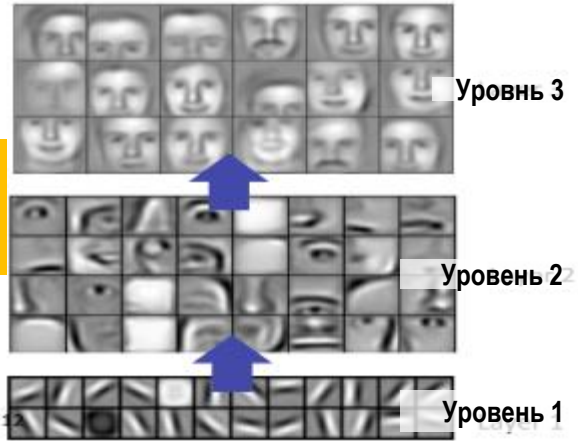
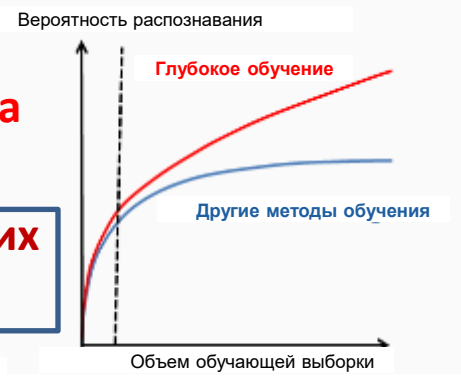
Исходный инструмент:
Перцептрон (MLP)

Сверточные нейронные сети в компьютерном зрении (первая волна технологической революции ИИ: 2011-2016)

Ключевое изобретение:
сверточный нейрон

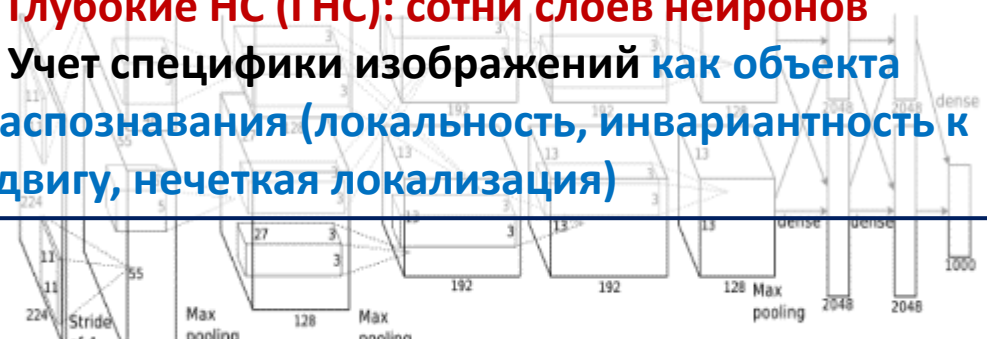
+ С 2011 г. - **распознавание образов на уровне человека или выше** (superhuman)

+ **Обучение на сверхбольших объемах данных**

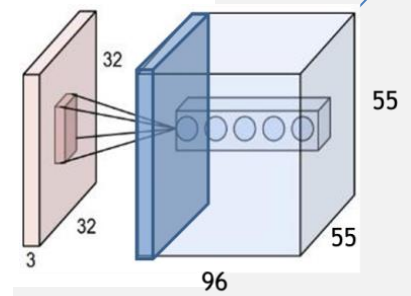
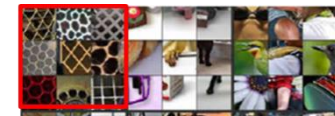
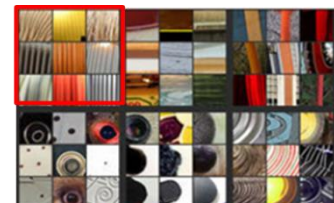
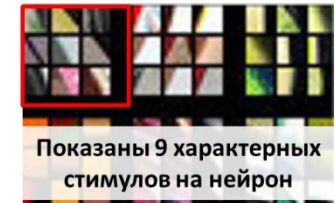


+ **Иерархическая обработка с повышением абстракции данных от уровня к уровню**

+ **Глубокие НС (ГНС): сотни слоев нейронов**
+ **Учет специфики изображений как объекта распознавания (локальность, инвариантность к сдвигу, нечеткая локализация)**



Типовая структура глубокой сверточной сети (CNN)



Работает локально в небольшой окрестности. Это естественно для изображений, где в любом масштабе есть локальные детали, которые нужно выделять и распознавать.

Какие элементы изображения распознают нейроны разных уровней: **чем выше слой сети, тем выше уровень абстракции**

Visualizing and Understanding Convolutional Networks, Matthew Zeiler and Rob Fergus, ECCV'14

2015-16: CNN решают все задачи компьютерного зрения и царят до 2020...

Источники эффективности

Революция трансформеров в анализе естественного языка (развитие трансформеров в области NLP, 2017-2023)

Deep Graph Embedding: глубокие сети на графах



На изображениях глубокие сети «снизу вверх» вычисляют структурные признаки окрестностей разного масштаба



Значит, и на графах нам нужна система подграфов разного масштаба

Опишем малые подграфы признаками, соберем из них признаки больших подграфов, и так – пока не опишем вектором признаков весь граф

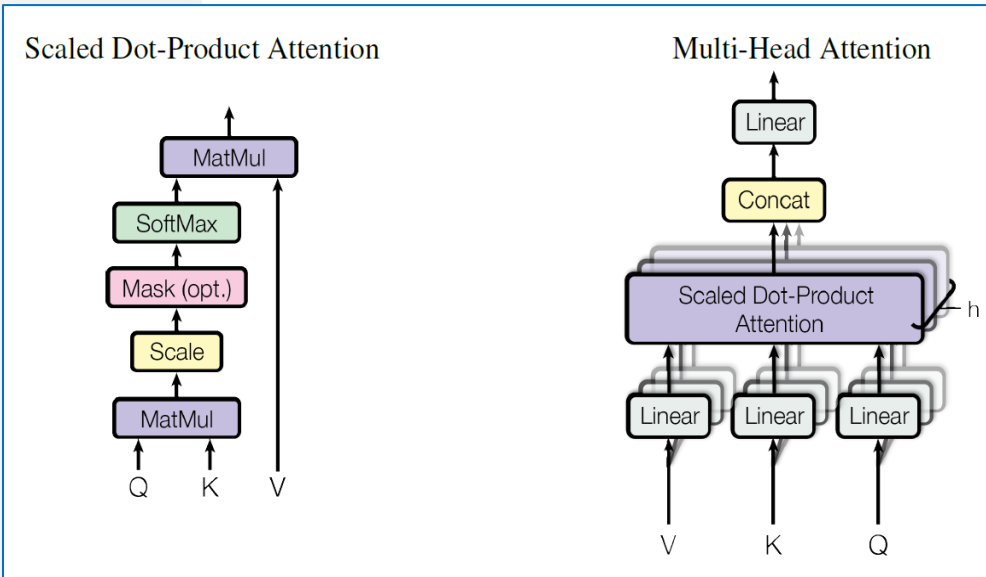
A Comprehensive Survey on Graph Neural Networks (Wu, Z. et al., 2019)

Рассмотрим эту задачу чуть подробнее. Глубокое обучение часто упрекают за отсутствие интересных математических задач и моделей. Мы постараемся показать, что это не совсем так...

В 2016-2019 гг. мы считали, что основным путем преодоления ограничений ГНС станут структурные ГНС на графах

Но мы ошиблись! Сегодня эту роль играет совсем другой тип ГНС, возникший в области NLP: сети-трансформеры, основанные на механизмах внимания

Attention Is All You Need (2017)



Q - Query. Множество векторов, для которых мы хотим посчитать attention.
K - Key. Множество векторов, с которыми мы ходим посчитать attention.
V - Value. Обычно совпадает с K.
Вычисляется скалярное произведение Q со всеми K, от результата берется softmax и с полученными весами перемноживается V.

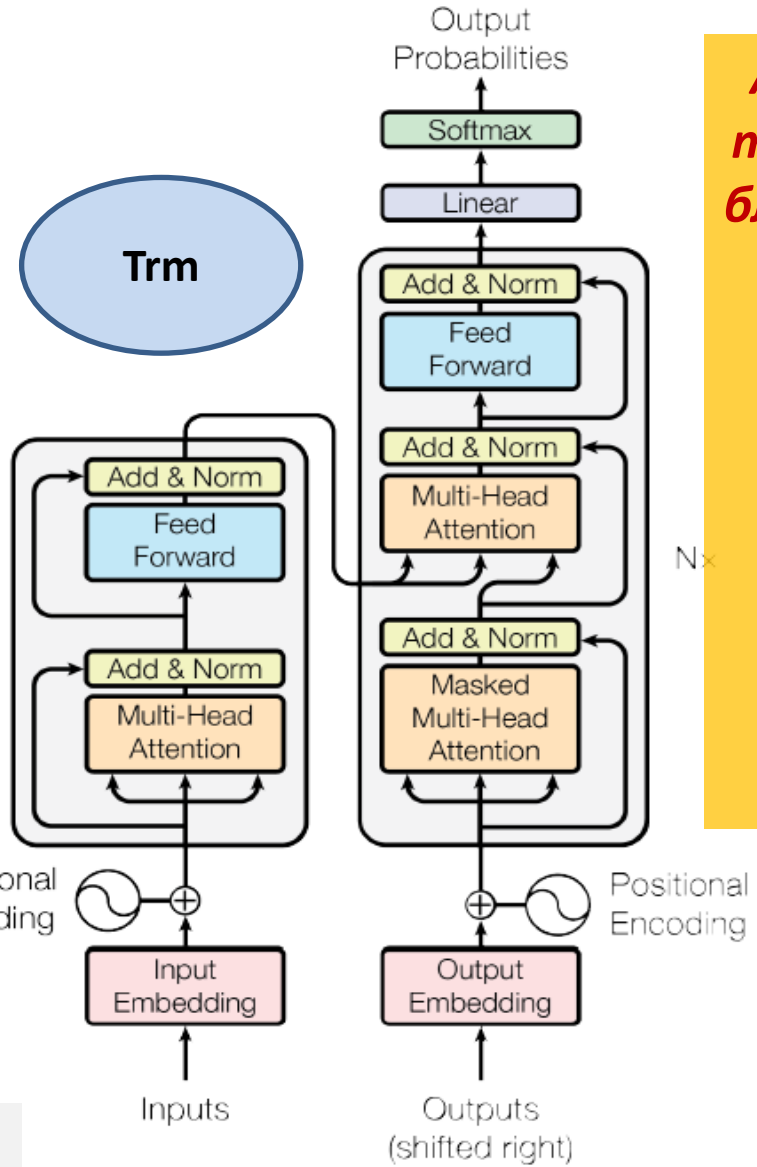
$$Q = XW^Q, \quad K = YW^K, \quad V = YW^V$$

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

An attention function can be described as **mapping a Query and a set of Key-Value pairs to an output**, where the query, keys, values, and output are all vectors.

Motivation: Learning long-range dependencies is a key challenge for conv layers, which is important for symbolic sequences.

В отличие от сверточного нейрона, модуль внимания работает нелокально, выделяя и используя смысловые связи между словами, далеко отстоящими в тексте



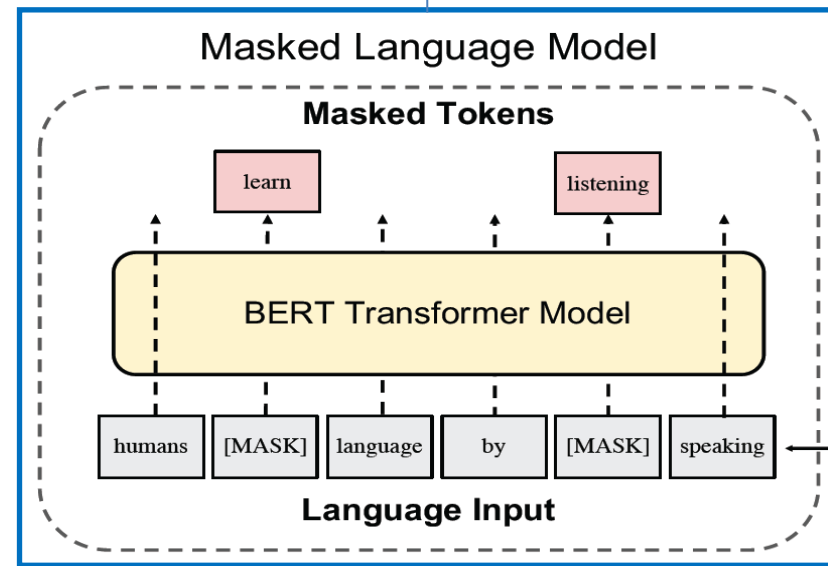
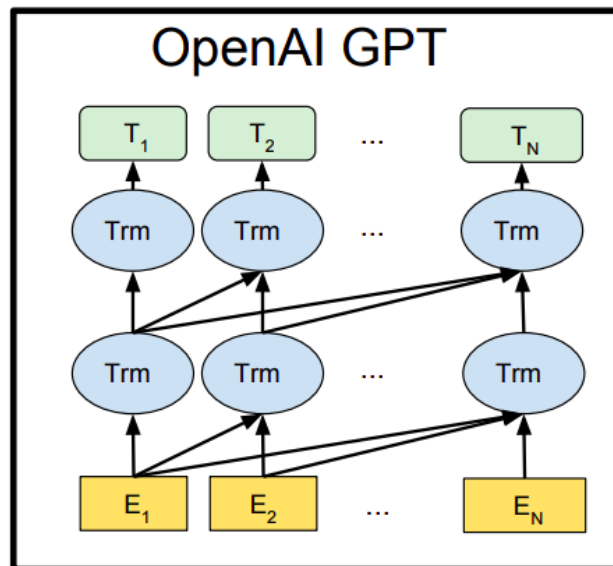
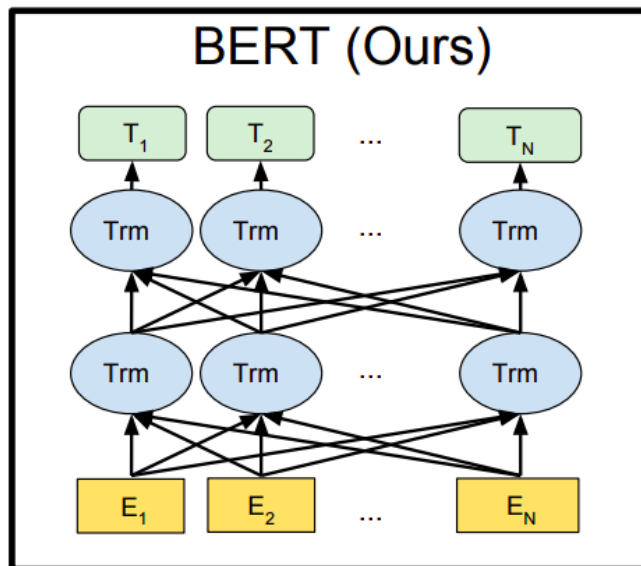
Архитектура трансформер, блоки которой состоят из модулей внимания, оказались ключом к решению самых сложных задач!

Figure 1: The Transformer - model architecture.

GPT и BERT (2018-19)

*Self-supervised переобучение
(без разметки данных!)*

Generative Pre-trained Transformer



Bidirectional Encoder Representations from Transformers (BERT)

BERT — это такой «трансформер», у которого увеличили число и размеры слоев, убрали декодирующую часть и научили **смотреть на контекст в обе стороны**. Внимание простого трансформера всегда направлено на токены слева от данного (слова справа заменяются особым словом [MASK], что обнуляет вес внимания). BERT же «маскирует» только то, что нужно предсказать, значит, внимание направлено на все токены входной последовательности — и слева, и справа.

<https://sysblok.ru/knowhow/kak-ustroena-nejroset-bert-ot-google/>

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
(Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, 2019)

GPT и BERT предобучаются на «замаскированном» тексте:
Часть слов в тексте маскируется, после чего сеть учится их предсказывать по оставшимся
GPT учится предсказывать продолжение текста
BERT - предсказать слово не в конце предложения, а где-то посередине любом

OpenAI GPT-3: “may be the biggest thing since bitcoin”

Параметры сети:

Размер сети 175 млрд. параметров (350 Гб fp16)

Контекстное окно: 2048 токена

96 трансформеров

96 self attention head

Размер батча: 3.2 млн

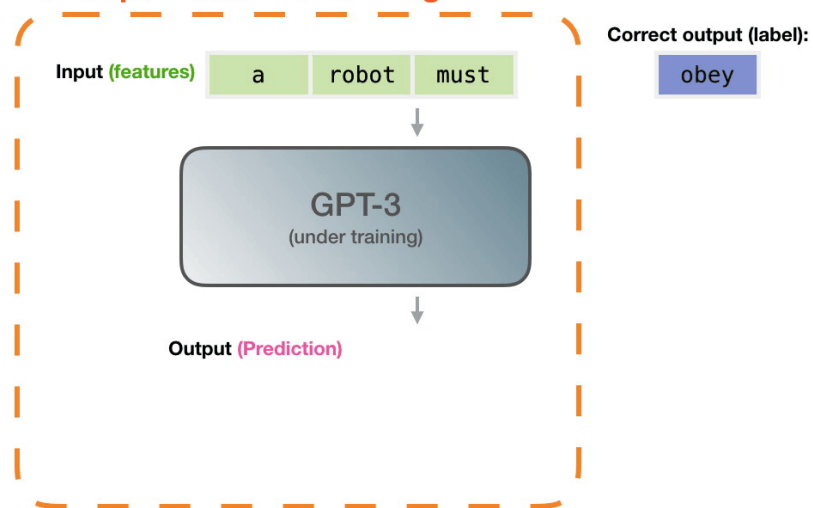
База данных:

The CommonCrawl data - 45 Тб

Прочие БД – 40%

Результаты сравнимые со state-of-the-art в 42 задачах NLP

Unsupervised Pre-training



Ключевая идея: обучение на огромной выборке на продолжение фразы (без учителя)

Суперкомпьютер:

10000 GPU Tesla V100

400 Гбит/сек

285 тыс. CPU ядер

Pytorch

Требует 355 лет и \$5 млн. при аренде GPU Cloud

Стало ясно, что возник качественно новый класс моделей: Большие языковые модели (LLM)

- GPT-3 может отвечать на вопросы на естественном языке
- GPT-3 умеет переводить с разных языков, писать статьи и т.д. вплоть до написания исходного кода программ

Не всегда понимает, чего от нее хотят

Много ошибок, почти все «на троечку»

**ХИТ ИИ
2020!**

LLM: Large Language Models (2021, 2022)

Модели становятся все больше!

- (English) **GPT-Neo** (2.7B), **GPT-J** (6B), **Jurassic-1** (7.5B/178B), **Gopher** (280B), **GPT-NeoX** (20B), **MT NLG** (530B), ...
- (Russian) **ruGPT-3** (13B)
- (Chinese) **CPM-2** (11B/198B* - MoE), **M6** (10B/100B), **Wu Dao 2.0** (1.75T*), **PangGu-α** (2.6B/13B/207B), **ERNIE 3.0** (10B) **Yuan 1.0** (245B), **ERNIE 3.0 Titan** (260B)
- (Korean) **HyperCLOVA** (204B)
- (Code) **OpenAI Codex** (12B), **Google's** (up to 137B)
- **ByT5** (up to 12.9B)
- **XLM-R XL/XXL** (3.5B/10.7B), **T-ULRv5 XL** (2.2B)
- **DeBERTa v1/v2** (1.5B), **v3** (300M)
- **Switch-C** (1.6T*), **GLaM** (1.2T*), **MoE LM by Meta** (1.1T*)
- **DALL·E** (12B)

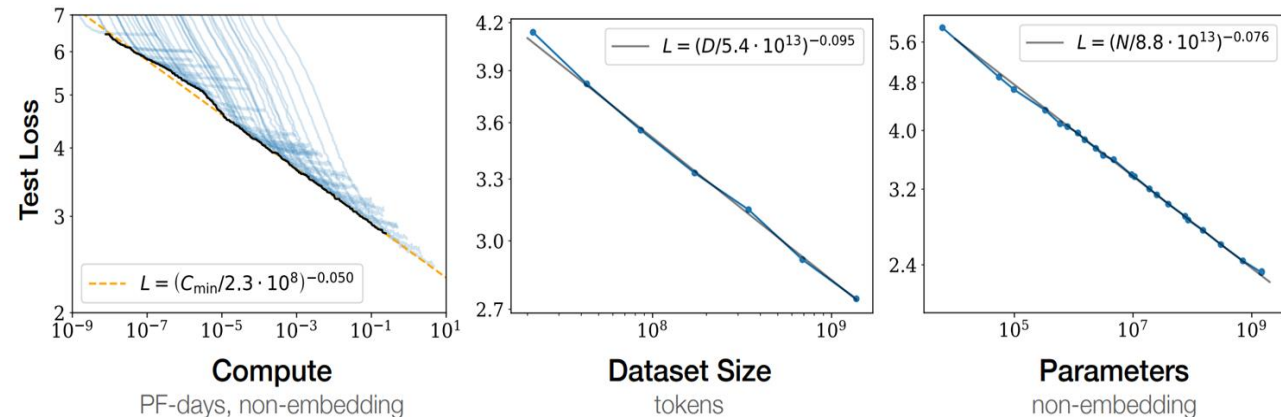


Figure 1 Language modeling performance improves smoothly as we increase the model size, dataset size, and amount of compute² used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

- «Мы исследовали эмпирические законы масштабирования для производительности языковой модели при кросс-энтропийной функции потерь.
- Потери масштабируются по степенному закону в зависимости от размера модели, размера набора данных и объема вычислений при обучении. При этом некоторые зависимости охватывают более семи порядков.

Много ошибок, почти все «на троечку»



Просто модель и база для обучения должны быть больше!

Приложения: Автоматическое программирование

Специализированные LLM для программирования и др. достигли уровня практического внедрения

Visual Studio Code interface showing a Go file named `runtime.go`. The code includes a `Run` struct and an `averageRuntimeInSeconds` function. A Copilot suggestion box is visible at the bottom left.

Codex (OpenAI)

<https://openai.com/blog/openai-codex/>

Другие известные модели:
PLBART
CodeT5
CodeGen
InCoder
SantaCoder...

AlphaCode (DeepMind)

<https://deepmind.com/blog/article/Competitive-programming-with-AlphaCode>

First AlphaCode reads the two phrases.

```
t=int(input())
for i in range(t):
    s=input()
    a=input()
    b=input()
    a.append(i)
    for j in t:
        b.append(j)
        a.reverse()
        b.reverse()
        c=[]
        while len(b)!=0 and len(a)!=0:
            if a[0]==b[0]:
                c.append(b.pop(0))
                a.pop(0)
            elif a[0]!=b[0] and len(a)!=1:
                a.pop(0)
                a.pop(0)
            elif a[0]!=b[0] and len(a)==1:
                a.pop(0)
        if len(b)==0:
            print("a")
        else:
            print("b")
```

Backspace deletes two letters. The letter you press backspace instead of, and the letter before it.

If the letters at the end of both phrases don't match, the last letter must be deleted. If they do match we can move onto the second last letter and repeat.

Много ошибок, почти все «на троечку»



Нельзя объять необъятное: если универсальные LLM много ошибаются, нужно делать специализированные LLM для конкретных областей приложений

Galactica: A Large Language Model for Science


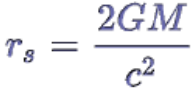
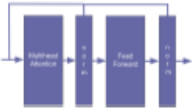
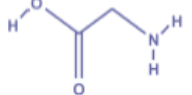

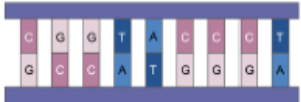
Modality	Entity	Sequence	
Text	Abell 370	Abell 370 is a cluster...	
L ^A T _E X	Schwarzschild radius	$r_{s} = \frac{2GM}{c^2}$	
Code	Transformer	<code>class Transformer(nn.Module)</code>	
SMILES	Glycine	<chem>C(C(=O)O)N</chem>	
AA Sequence	Collagen α -1(II) chain	MIRLGAPQTL..	
DNA Sequence	Human genome	CGGTACCCTC..	

Table 1: Tokenizing Nature. Galactica trains on text sequences that represent scientific phenomena.

We introduce a new large language model called **Galactica (GAL)** for automatically organizing science. Galactica is trained on a large and curated corpus of humanity's scientific knowledge. This includes over **48 million papers, textbooks and lecture notes**, millions of compounds and proteins, scientific websites, encyclopedias and more

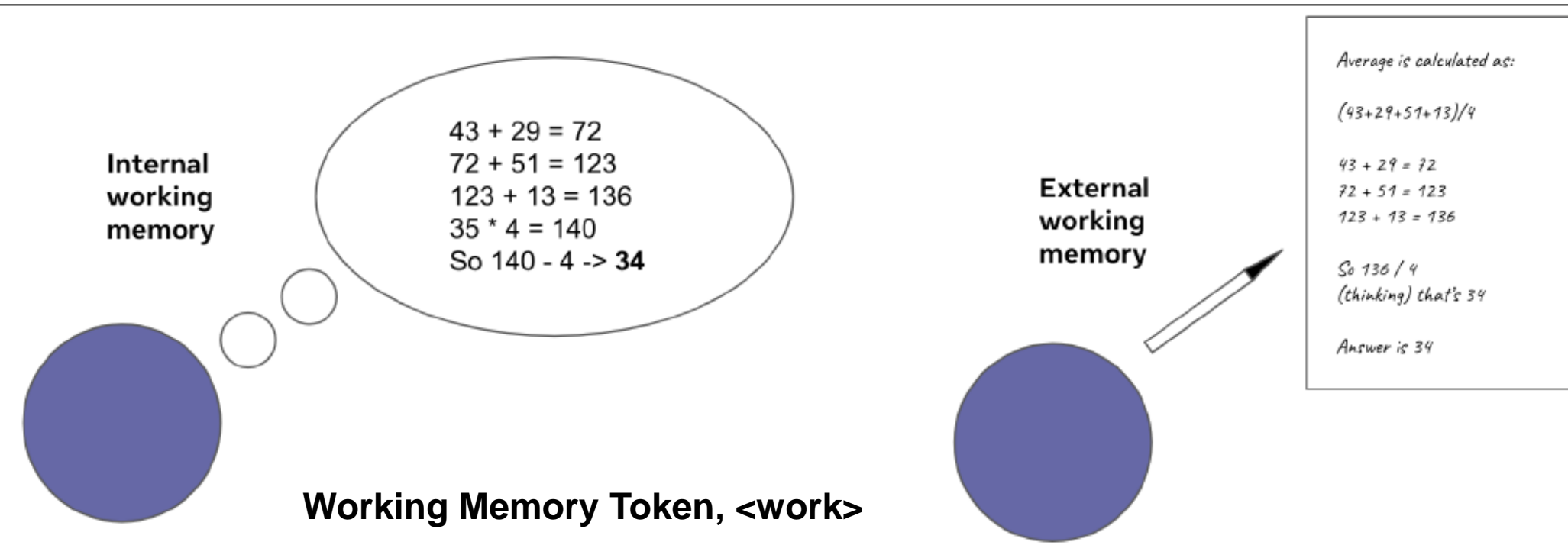
Total dataset size = 106 billion tokens

Data source	Documents	Tokens	Token %
Papers	48 million	88 billion	83.0%
Code	2 million	7 billion	6.9%
Reference Material	8 million	7 billion	6.5%
Knowledge Bases	2 million	2 billion	2.0%
Filtered CommonCrawl	0.9 million	1 billion	1.0%
Prompts	1.3 million	0.4 billion	0.3%
Other	0.02 million	0.2 billion	0.2%

Много ошибок, почти все «на троечку»

Специализация может быть не очень узкой (LLM для целой области)

Galactica: A Large Language Model for Science



Transformer-based architectures lack an explicit working memory capability, which means a single-forward pass has limited efficacy. This is problematic for tasks that require multiple steps of computation. A current workaround is using a Transformer's output context as an external working memory to read from and write to. This is seen in recent work on **chain-of-thought prompting** (Wei et al., 2022; Suzgun et al., 2022).

Our solution is a **working memory token** we call <work>.

Много ошибок, почти все «на троечку»

Специализация
может быть не
очень узкой
(LLM для целой
области)

Galactica: A Large Language Model for Science

Question: A needle 35 mm long rests on a water surface at 20°C. What force over and above the needle's weight is required to lift the needle from contact with the water surface? $\sigma = 0.0728\text{m}$.

<work>

$$\begin{aligned}\sigma &= 0.0728 \text{ N/m} \\ \sigma &= F/L \\ 0.0728 &= F/(2 \times 0.035) \\ F &= 0.0728(2 \times 0.035)\end{aligned}$$

calculate.py

```
'''  
f = 0.0728*(2*0.035)
```

```
with open("output.txt", "w") as file:  
    file.write(str(round(f, 5)))
```

```
'''
```

«run: "calculate.py"»

«read: "output.txt"»

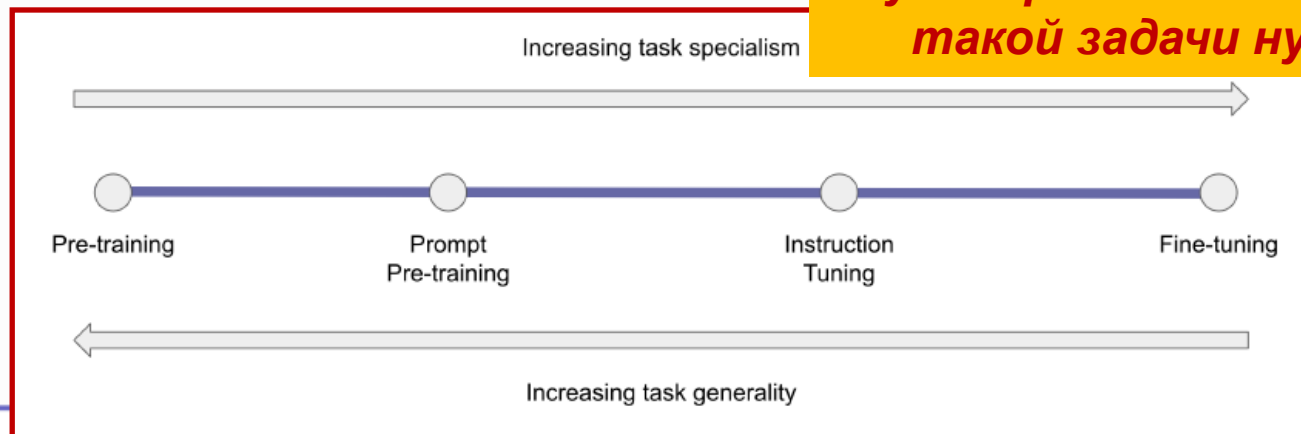
0.0051

</work>

Answer: $F = 0.0051 \text{ N}$

Where a computation is performed that a human could not do internally, we offload by writing and executing a Python script. Importantly, we do not have to turn this on, and **the model can also predict the output from running a program**. For our experiments, we did not find the need to turn Python offloading on, and leave this aspect to future work.

Проблема создания таких специализированных приложений в том, что мы уходим от универсальности трансформеров – для каждой такой задачи нужно придумывать свои трюки



Много ошибок, почти все «на троечку»

Специализация может быть не очень узкой (LLM для целой области)

Retrieval-based models (2021+)

- **EMDR²** (“End-to-End Training of Multi-Document Reader and Retriever for Open-Domain Question Answering”, <https://arxiv.org/abs/2106.05346>)
- **SPALM** (“Adaptive Semiparametric Language Models”, <https://arxiv.org/abs/2102.02557>)
- **BlenderBot 2.0** (“Internet-Augmented Dialogue Generation”, <https://arxiv.org/abs/2107.07566>)
- **DeepMind RETRO** (“Improving language models by retrieving from trillions of tokens”, <https://arxiv.org/abs/2112.04426>) **2021**
- **OpenAI WebGPT** (“WebGPT: Browser-assisted question-answering with human feedback”, <https://arxiv.org/abs/2112.09332>)
- **Toolformer** (Toolformer: Language Models Can Teach Themselves to Use Tools) **2023** <https://arxiv.org/abs/2302.04761>

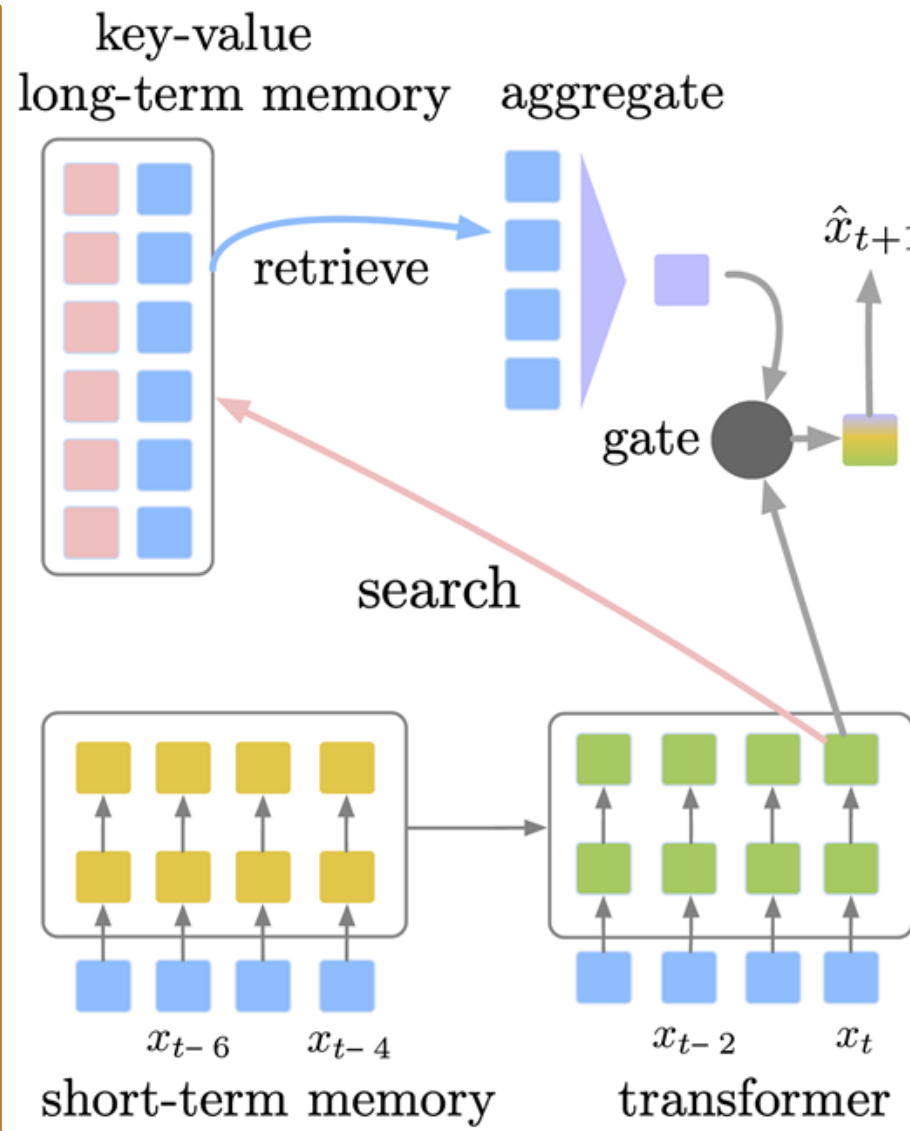
Много ошибок, почти все «на троечку»



With a 2 trillion token database, RETRO (Retrieval-Enhanced Transformer) obtains **comparable performance to GPT-3** and Jurassic-1 on the Pile, despite **using 25× fewer parameters.**

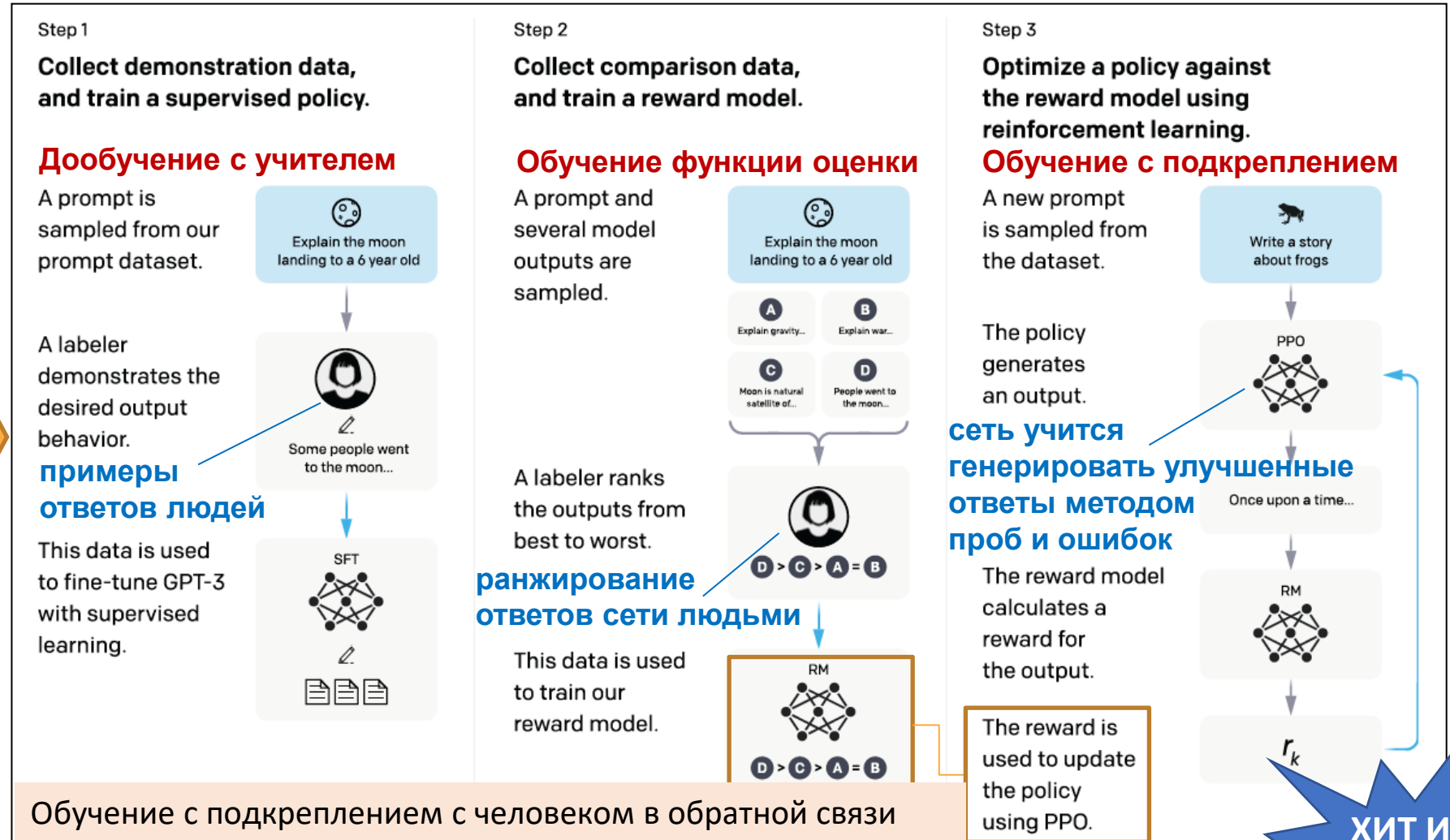
With RETRO the model is not limited to the data seen during training– it has access to the entire training dataset through the **retrieval mechanism.**

Автоматический поиск данных в коллекциях и открытых источниках для ответов на вопросы



ChatGPT (InstructGPT): как добиться от GPT полезных и релевантных ответов при помощи обучения с подкреплением

We want language models to be **helpful** (they should help the user solve their task), **honest** (they shouldn't fabricate information or mislead the user), **harmless** (they should not cause physical, psychological, or social harm to people or the environment).



Не всегда понимает, чего от нее хотят

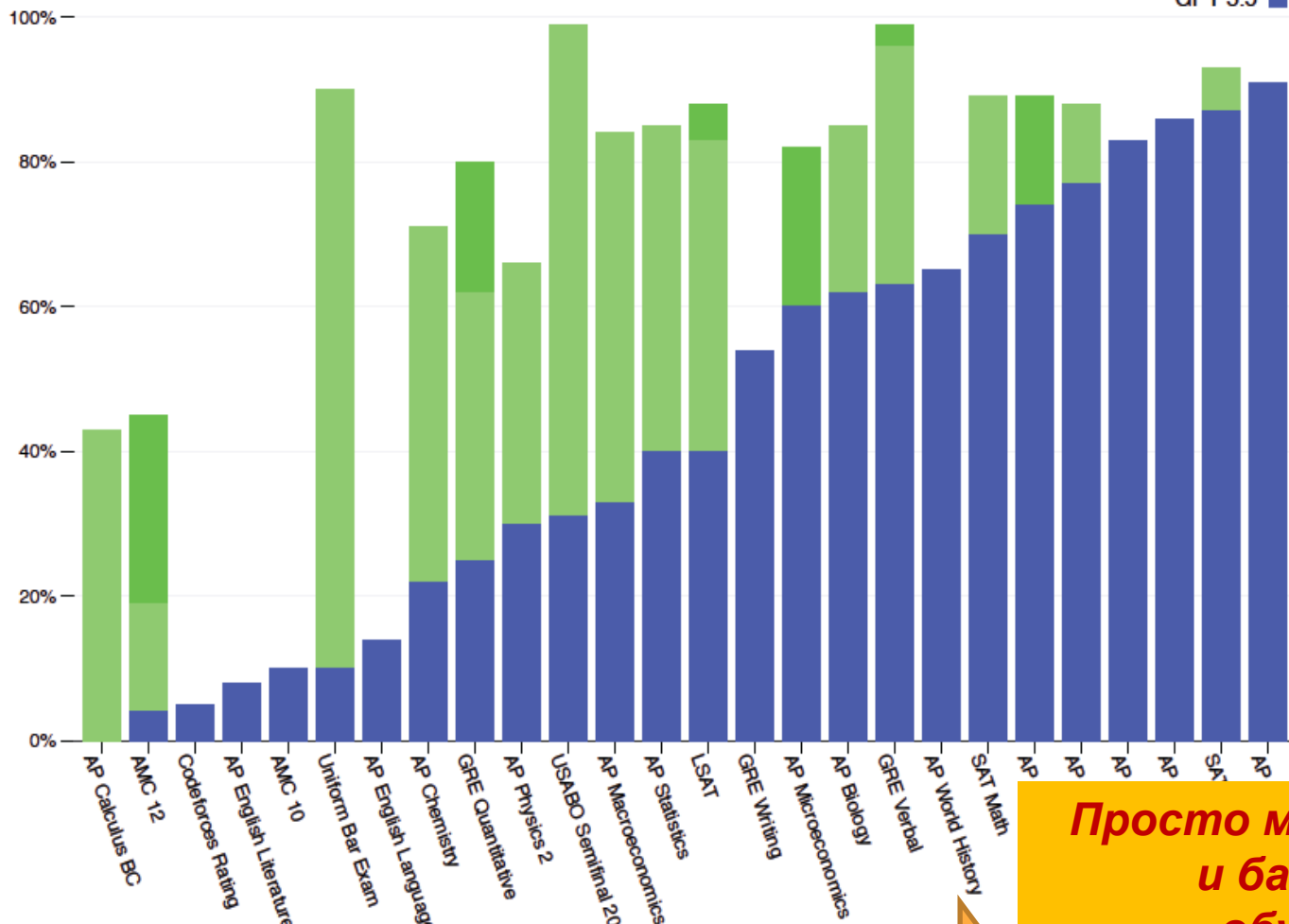
Reinforcement Learning from Human Feedback (RLHF)



GPT-4 (март 2023): намного «умнее» предыдущей версии ChatGPT

Exam results (ordered by GPT 3.5 performance)

Estimated percentile lower bound (among test takers)



Много ошибок, почти все «на троечку»

Просто модель и база для обучения должны быть ГОРАЗДО больше!

Exam
Uniform Bar Exam (MBE+MEE+MPT)
LSAT
SAT Evidence-Based Reading & Writing
SAT Math
Graduate Record Examination (GRE) Quantitative
Graduate Record Examination (GRE) Verbal
Graduate Record Examination (GRE) Writing
USABO Semifinal Exam 2020
USNCO Local Section Exam 2022
Medical Knowledge Self-Assessment Program
Codeforces Rating
AP Art History
AP Biology
AP Calculus BC
AP Chemistry
AP English Language and Composition
AP English Literature and Composition
AP Environmental Science
AP Macroeconomics
AP Microeconomics
AP Physics 2
AP Psychology
AP Statistics
AP US Government
AP US History
AP World History
AMC 10
AMC 12
Introductory Sommelier (theory knowledge)
Certified Sommelier (theory knowledge)
Advanced Sommelier (theory knowledge)
Leetcode (easy)
Leetcode (medium)
Leetcode (hard)

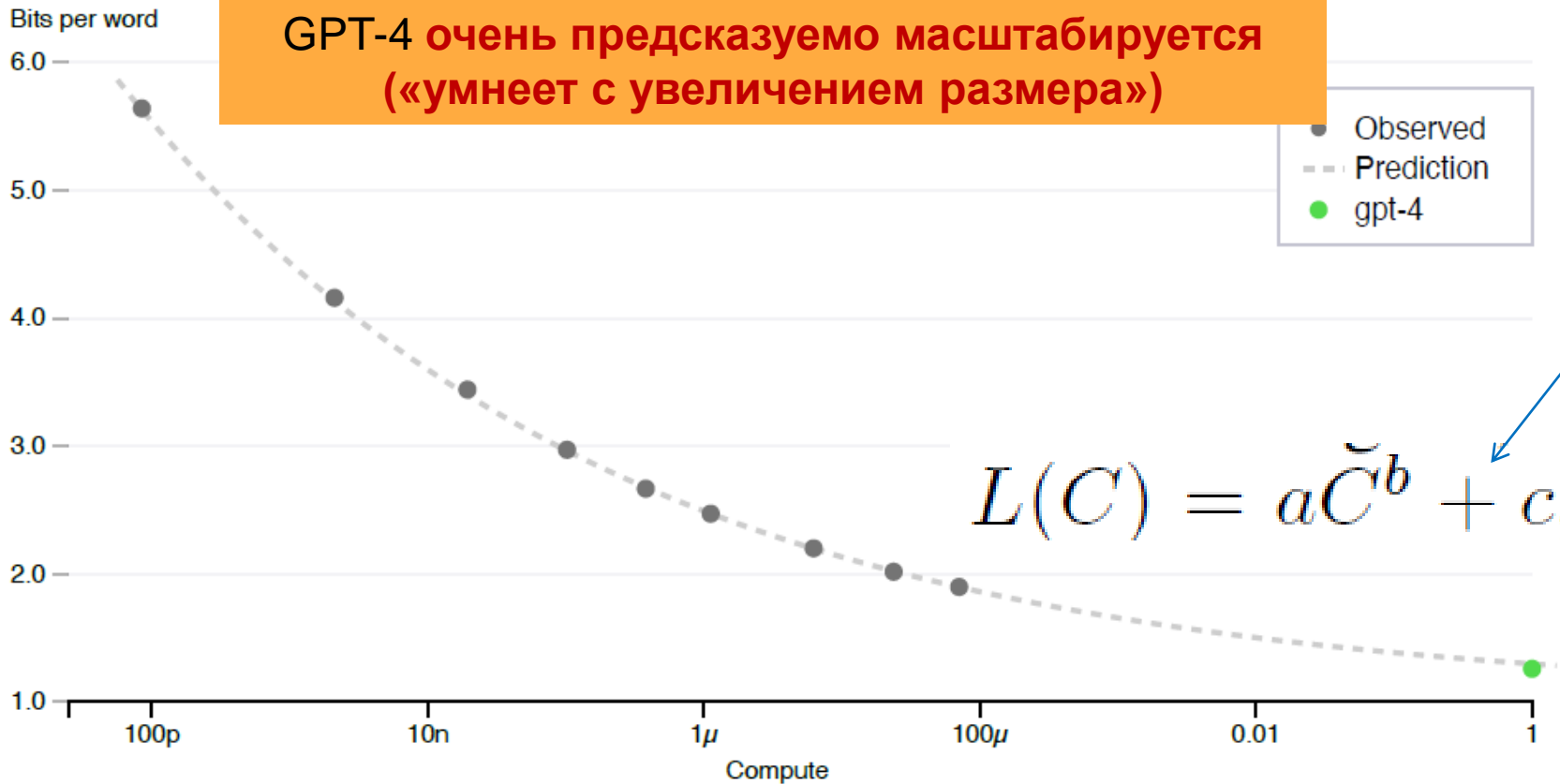
GPT performance on academic and professional exams

80% проф. тестов GPT-4 проходит! Т.е. она готова заменить людей в 80% профессий, где требуется высшее образование

ХИТ ИИ 2023!

GPT-4 (март 2023): «ум» зависит от размера «мозга» и «опыта»

OpenAI codebase next word prediction



The final loss of properly-trained large language models is thought to be well approximated by power laws in the amount of compute used to train the model

We predicted GPT-4's final loss on our internal codebase (not part of the training set), from models trained using the same methodology but using at most 10,000x less compute than GPT-4.

A large focus of the GPT-4 project was building a deep learning stack that scales predictably. The primary reason is that for very large training runs like GPT-4, it is not feasible to do extensive model-specific tuning.

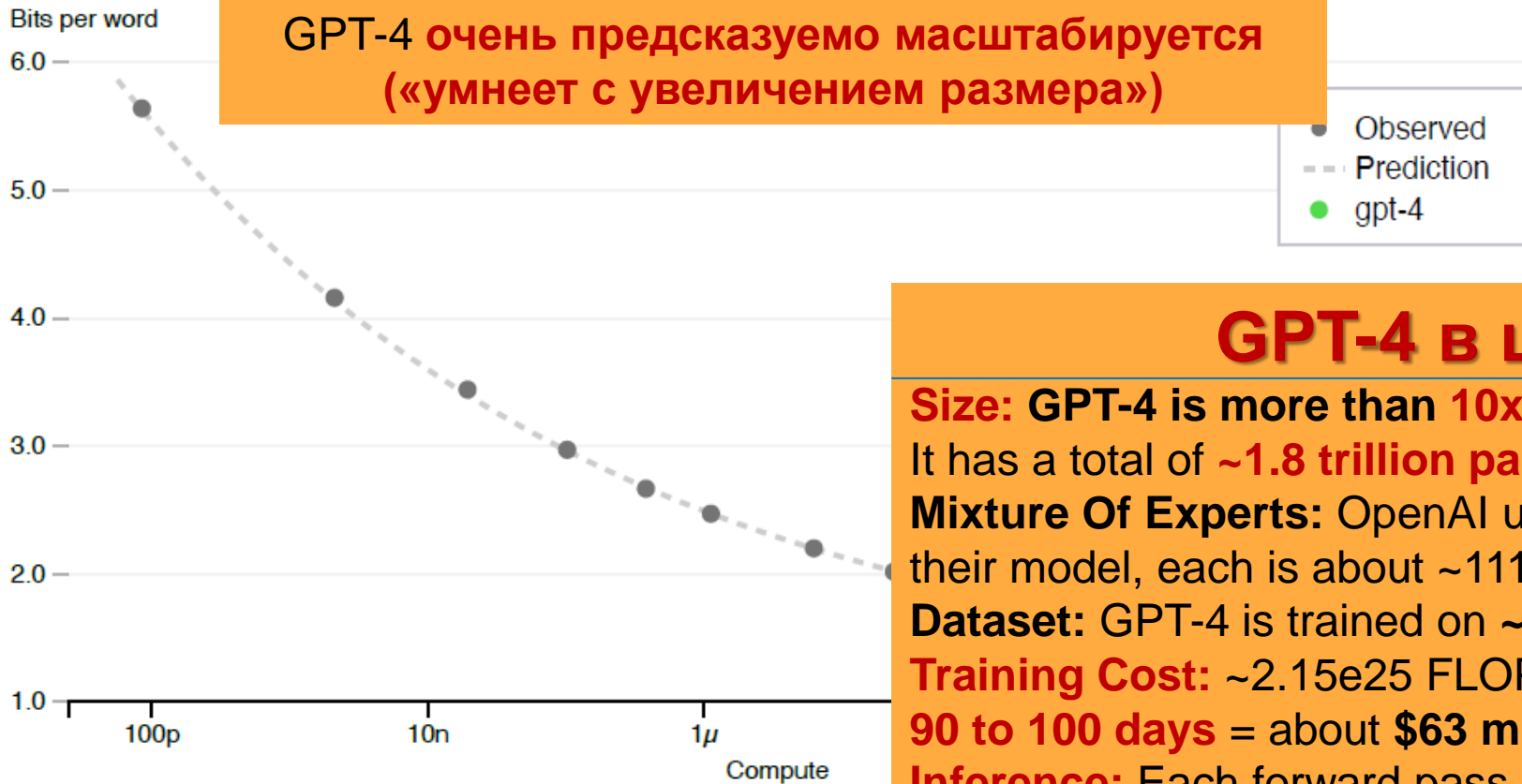
GPT-4 Technical Report, OpenAI, 2023

ВАЖНО: Зависимость качества от размера сети, объема данных и времени вычислений формульная и хорошо предсказуемая на всем диапазоне масштабов!



GPT-4 (март 2023): «ум» зависит от размера «мозга» и «опыта»

OpenAI codebase next word prediction



The final loss of properly-trained large language models is thought to be well approximated by power laws in the amount of compute used to train the model.

GPT-4 в цифрах

Size: GPT-4 is more than 10x the size of GPT-3.

It has a total of ~1.8 trillion parameters in 120 layers.

Mixture Of Experts: OpenAI utilizes 16 experts within their model, each is about ~111B parameters for MLP.

Dataset: GPT-4 is trained on ~13T (10^{12}) tokens.

Training Cost: ~2.15e25 FLOPS, on ~25,000 A100s for 90 to 100 days = about \$63 million.

Inference: Each forward pass inference (generation of 1 token) utilizes ~280B parameters and ~560 TFLOPs.

Inference Cost GPT-4 costs 3x that of the 175B parameter Davinchi. About \$0.0049 cents per 1k tokens for 128 A100s to inference GPT-4.

<https://threadreaderapp.com/thread/1678545170508267522.html>

A large focus of the GPT-4 project was building a deep learning stack that scales predictably. The primary reason is that for very large training runs like GPT-4, it is not feasible to do extensive model-specific tuning.

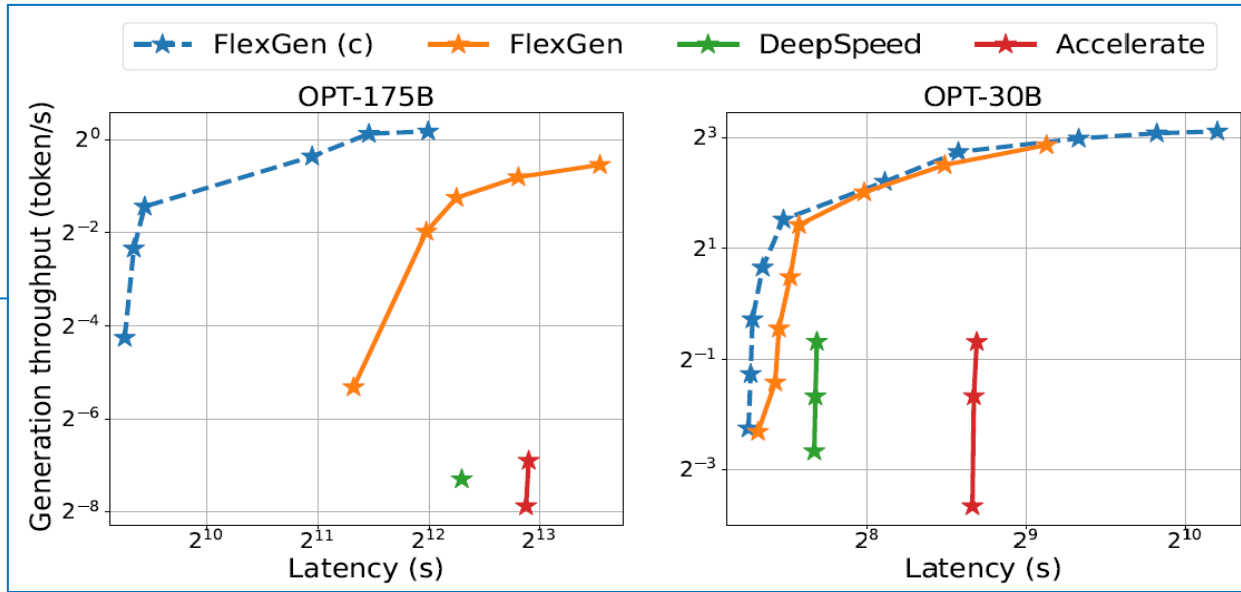
T-4's internal part of the models same

t compute

ИИ 3!

Есть ли перспектива использования LLM в бортовом ИИ?

LLM может работать быстрее



175B параметров = GPT-3

FlexGen (март 2023): выполнение 175B LLM на одном 16GB GPU <https://github.com/FMInference/FlexGen>

Обучение LLM простым смертным недоступно. Однако использование в своих конечных и даже бортовых решениях вполне возможно! Также доступно дообучение (не на борту).

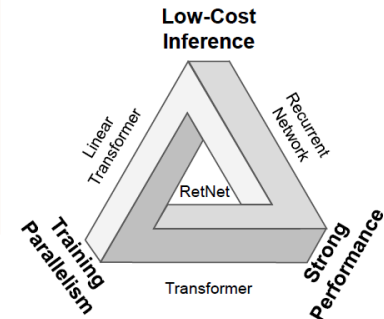
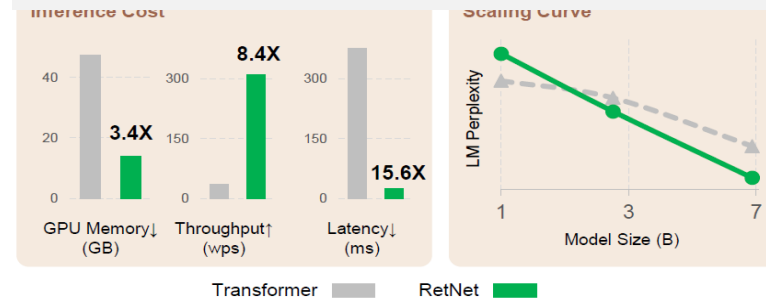
High-throughput Generative Inference of Large Language Models with a Single GPU, 2023

LLM может иметь другую архитектуру

Направления исследований:

- linearized attention [KVPF20]
- recurrent models for efficient inference while sacrificing training parallelism [PAA+23]
- replacing attention with other mechanisms, such as S4 [GGR21], and its variants [DFS+22, PMN+23].

Retentive Network (RETNET) - LLM, achieving training parallelism, low-cost inference, and performance.



[KVPF20] Transformers are RNNs: Fast autoregressive transformers with linear attention.

[PMN+23] Towards larger convolutional language models.

[GGR21] Efficiently modeling long sequences with structured state spaces

Retentive Network: A Successor to Transformer for Large Language Models, Sun et al., 2023

Успехи есть!

Трансформеры и CNN
в задачах зрения и multi-modal
(развитие трансформеров и CNN
в области CV, 2020-2023)

Трансформеры идут в CV! (2020+)

Классификация ImageNet

Rank	Model	Accuracy	Params	Year	Architecture
1	VIT-G/14	90.45%	1843M	2021	Scaling Vision Transformers
	VIT-MoE-15B			2021	Scaling Vision with Mixture of
				2021	Pseudo Labels
				2021	Pseudo Labels
				2021	Performance at Scale ImageNet

Обнаружение MS COCO

Rank	Model	AP50	AP75	AP	APM	APL	Year	Architecture	
1	DyHead (Swin-L, multi scale, self-training)	60.6	78.5	66.6	43.9	64.0	74.2	2021	Dynamic Head: Unifying Object Detection Heads with Attention
2	DyHead (Swin-L, multi scale)	58.7	77.1	64.5	41.7	62.0	72.8	2021	Dynamic Head: Unifying Object Detection
								2021	
								2021	
								2021	
								2020	

Сегментация ADE20K

Rank	Model	mIoU	Year	Architecture
1	CSWin-L (UperNet, ImageNet-22k pretrain)	55.2%	2021	CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows
			2021	with Patch
			2021	ed Image
			2021	mer for ion
			2021	erarchical using Shifted

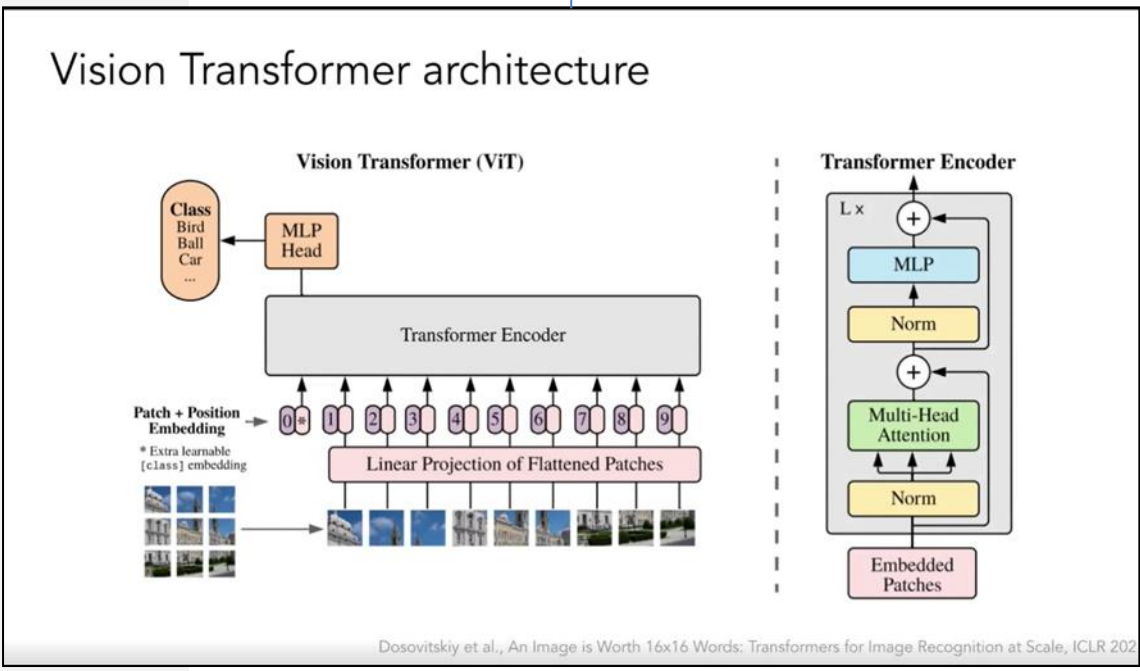
Сегментация(instance) MS COCO

Rank	Model	AP50	Year	Architecture
1	Swin-L (HTC++, multi scale)	51.1	2021	Swin Transformer: Hierarchical Vision Transformer using Shifted Windows
			2021	Swin Transformer: Hierarchical Vision Transformer using Shifted Windows
			2021	Instances as Queries
			2020	Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation
			2020	DetectorRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution

Базовая идея применения трансформеров к задачам зрения: вместо текстовых токенов - фрагменты изображений

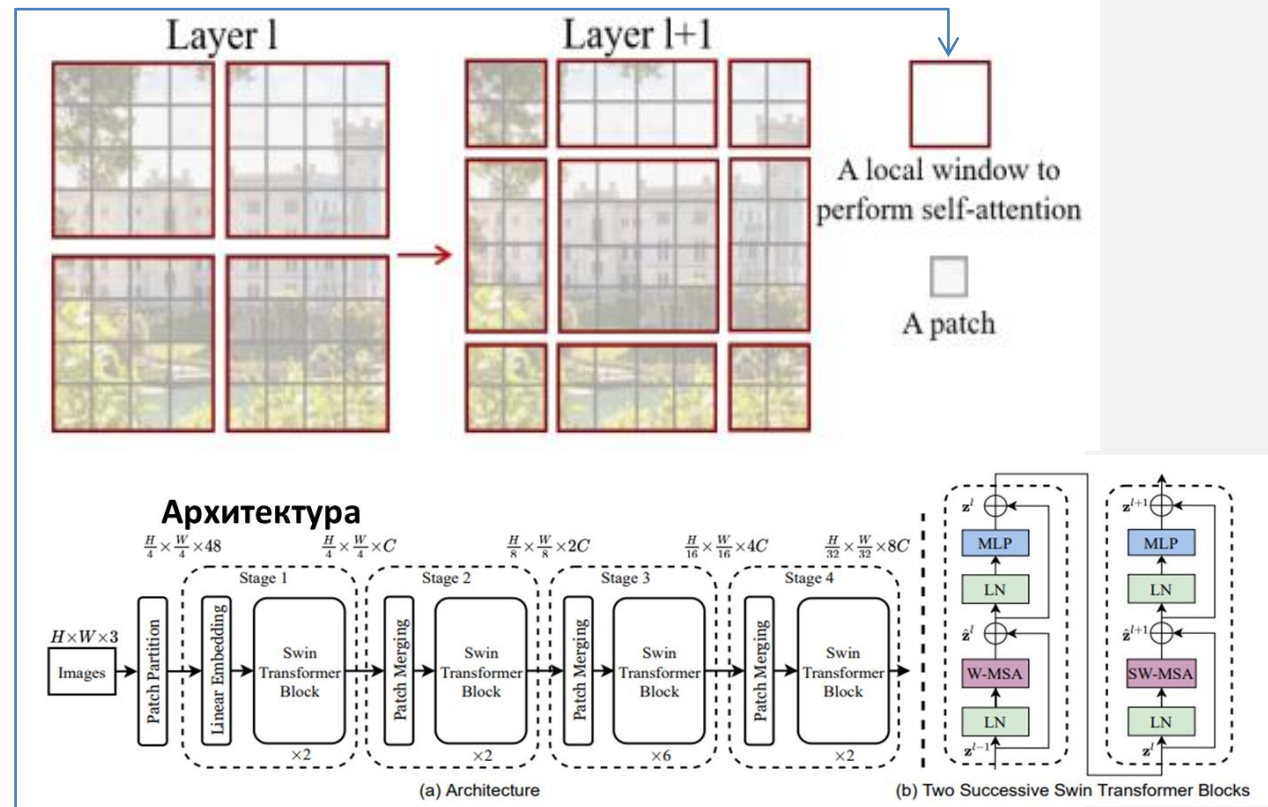
Трансформеры идут в CV! (2020+)

Vision Transformer (2020)



**Изображение как «текст»:
все токены сравниваются попарно,
отсюда квадратичные вычисления**

Shifted Windows Swin Transformer (2021)



Трансформеры гораздо медленнее сверточных сетей из-за квадратичных вычислений, но если их блоки применять локально, то скорость существенно улучшается!

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby

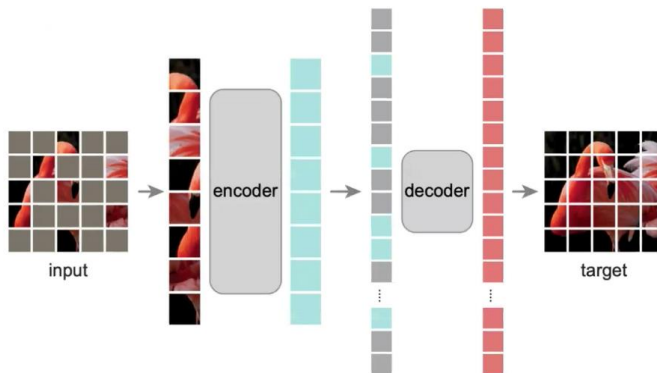
MAE: Masked Autoencoders As Scalable Vision Learners (2021)

Self-Supervised Learning

MAE: Masked Autoencoders Are Scalable Vision Learners

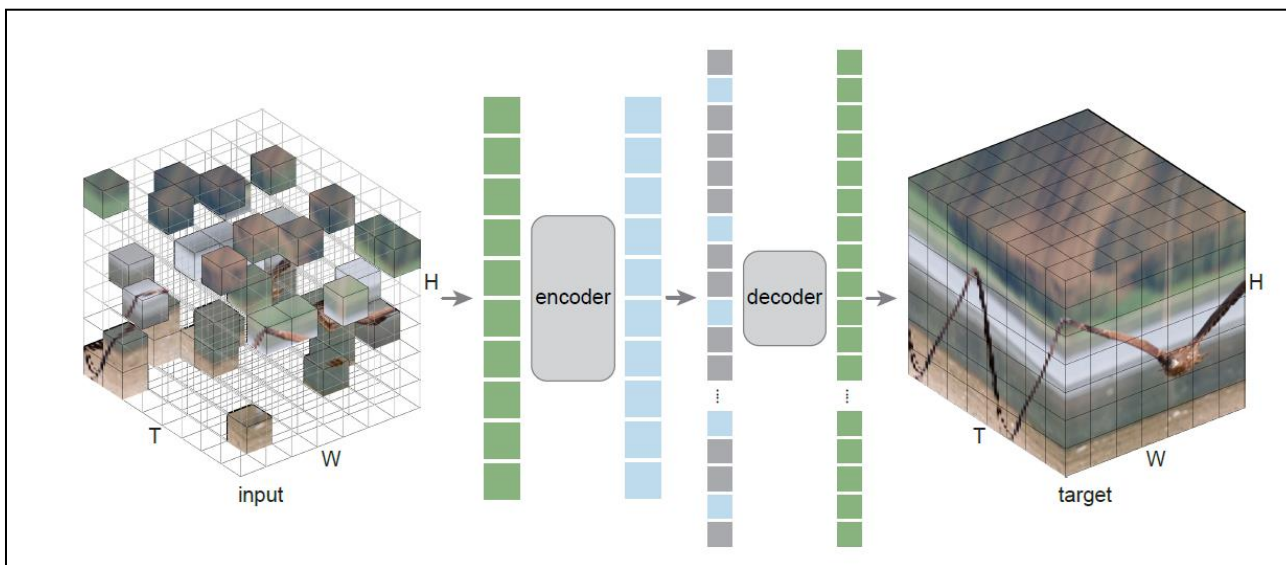
Better BERT-style training with a few tricks:

- Only encode non-masked tokens
- Use a decoder to predict missing patches

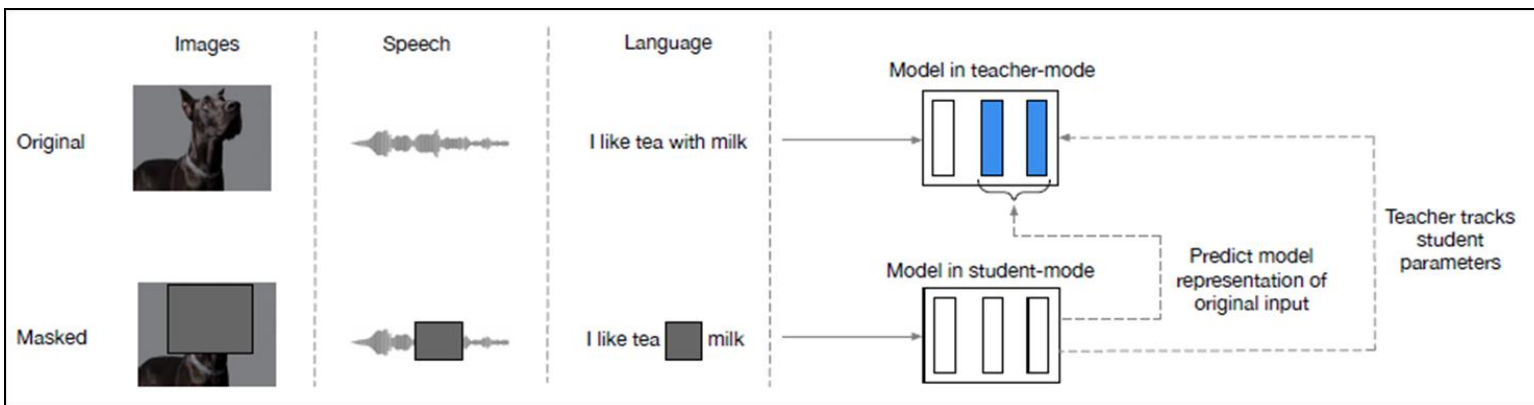


Simple autoencoding of RGB pixel values works extremely well!

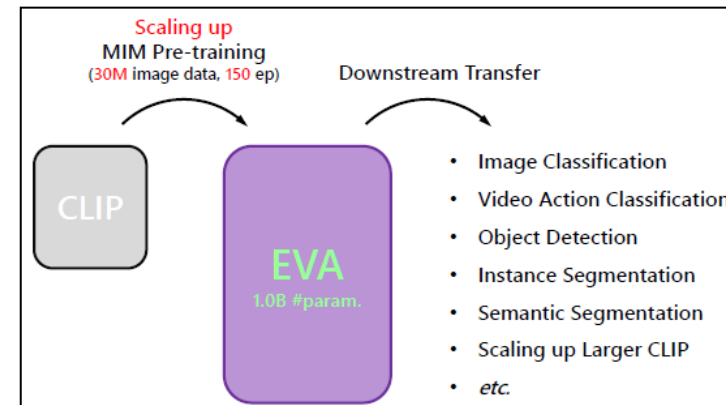
Kaiming He et al., Masked Autoencoders Are Scalable Vision Learners, arXiv 2021



MAE: Masked Autoencoders As Spatiotemporal Learners, Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, Kaiming He, Meta AI, FAIR, 2022



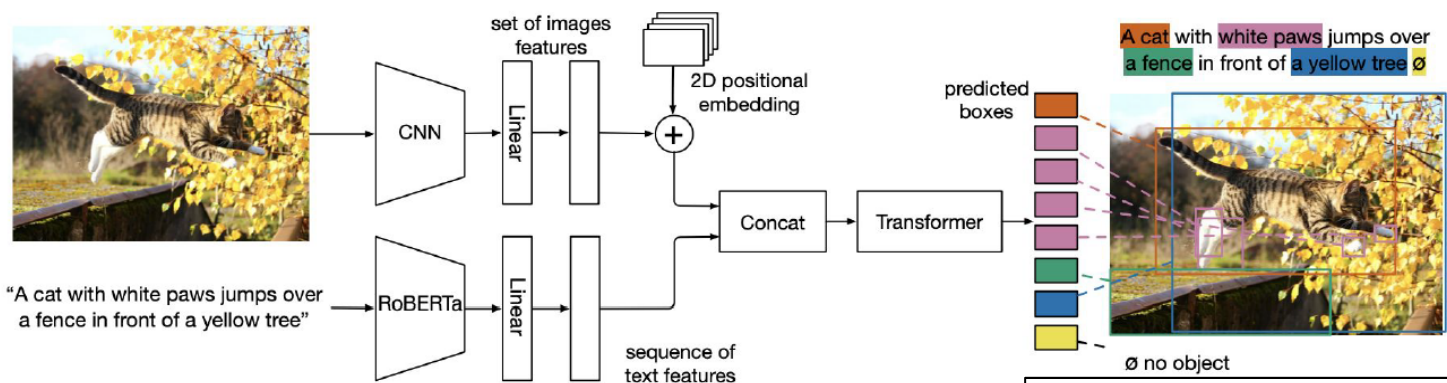
data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language, MetaAI, 2022



EVA: Exploring the Limits of Masked Visual Representation Learning at Scale, Fang et al., 2022

Сети GPT и BERT обучают на восстановление «замаскированных» фрагментов текста. Оказалось для изображений, видео и других типов данных это тоже работает!

Multimodal DETR (M-DETR)



Анализ визуальных данных по текстовому описанию на естественном языке

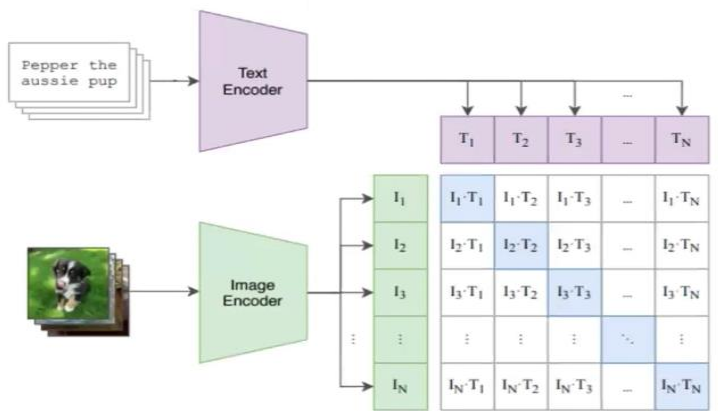
Генерация визуальных данных по текстовому описанию на естественном языке

Self-Attention for Vision

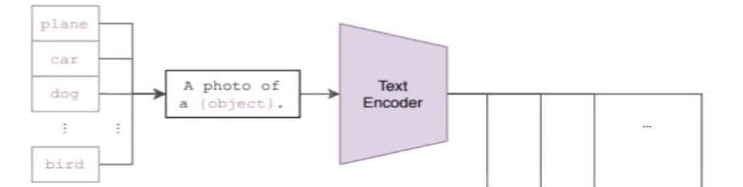
Ashish Vaswani¹, Prajit Ramachandran¹, and Aravind Srinivas²
¹ Google Research, ² UC Berkeley

CLIP: Visual representations from image-and-text data

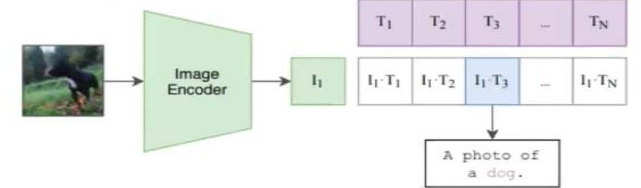
(1) Contrastive pre-training



(2) Create dataset classifier from label text



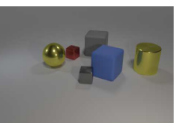
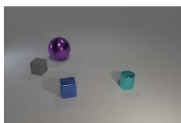
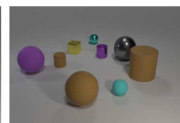
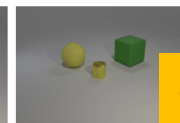

(3) Use for zero-shot prediction



Текстовые и визуальные данные переводятся в единое представление и потом обрабатываются совместно!

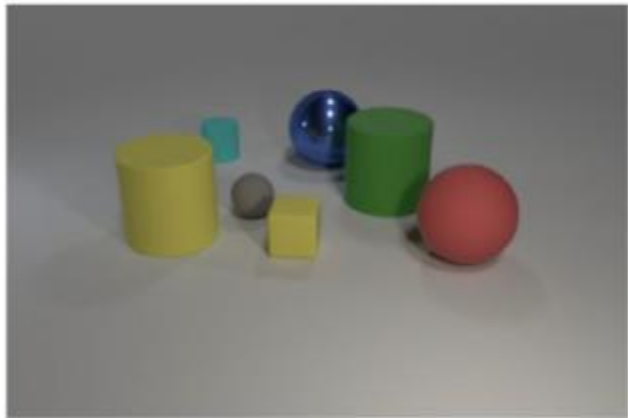
Совместная обработка сигнальной и символьной информации

M-DETR

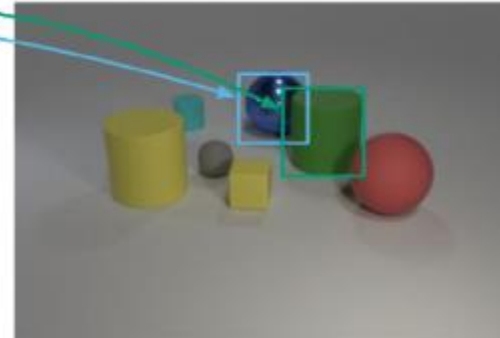
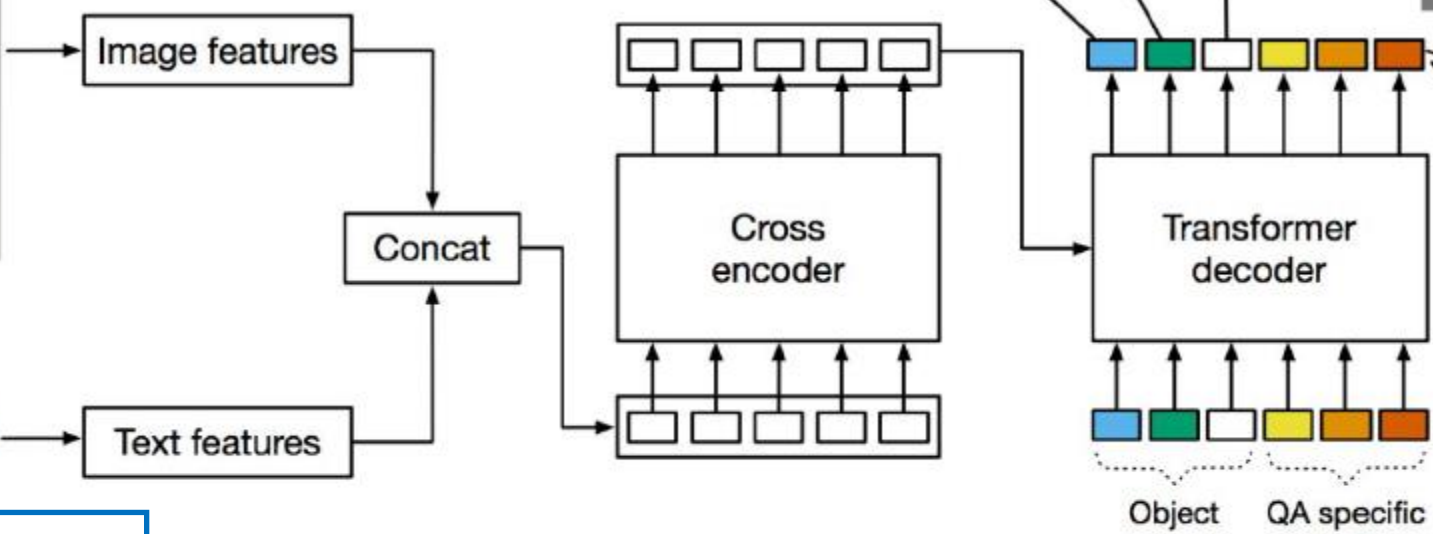
				
Q: Is there a <u>blue box</u> in the <u>items</u> ? A: yes	Q: What shape object is <u>farthest</u> right? A: cylinder	Q: Are <u>all</u> the balls small? A: no	Q: Is the green block to the right of the <u>yellow sphere</u> ? A: yes	Q: Two items share a color, a material, and a shape; what is the size of the <u>tightmost</u> of those items? A: large
Predicted Program: exist filter_shape[cube] filter_color[blue] scene	Predicted Program: query_shape unique relate[right] unique filter_shape[cylinder] filter_color[blue] scene	Predicted Program: equal_size query_size unique filter_shape[sphere] scene query_size unique filter_shape[sphere] filter_size[small] scene	Predicted Program: exist filter_shape[cube] filter_color[green] relate[right] unique filter_shape[sphere] filter_color[yellow] scene	Predicted Program: count filter_shape[cube] same_material unique filter_shape[cylinder] scene
Predicted Answer: ✓ yes	Predicted Answer: ✓ cylinder	Predicted Answer: ✓ no	Predicted Answer: ✓ yes	Predicted Answer: ✗ 0

2017: ИИ-1 встречается с ИИ-2!!!

Ответы на визуальные вопросы, требующие рассуждений: «визуальный тест Тьюринга»



“What is the color of the sphere behind the green cylinder?”



Self-Attention for Vision

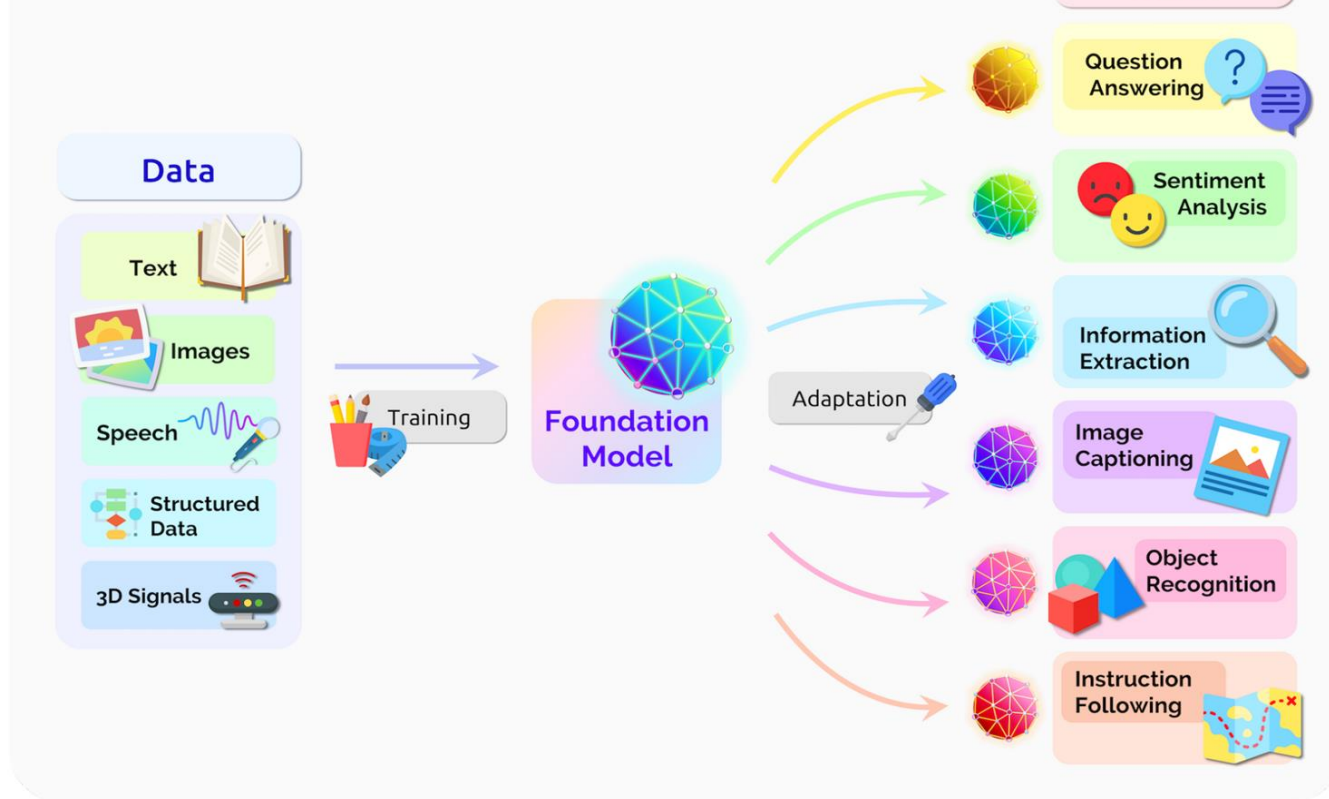
Ashish Vaswani¹, Prajit Ramachandran¹, and Aravind Srinivas²

¹ Google Research, ² UC Berkeley

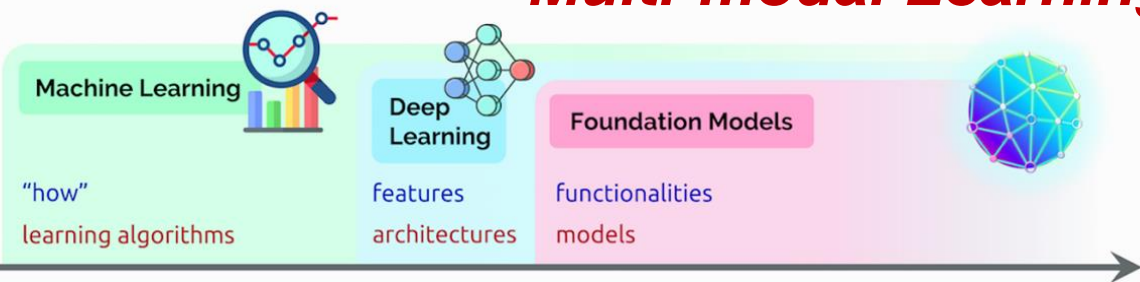
2021: Раньше казалось, что здесь нужны базы знаний и логическое программирование, теперь это делают трансформеры!

Foundation Models (2020+)

Идея: обучить одну модель на большом объеме данных, а потом сразу применять ее для множества приложений



Emergence of...
Homogenization of...



AI is undergoing a paradigm shift with the rise of models (e.g., BERT, DALL-E, GPT-3) trained on broad data (generally using self-supervision at scale) that can be adapted to a wide range of downstream tasks. **We call these models foundation models** to underscore their critically central yet incomplete character.

This report provides a thorough account of the opportunities and risks of foundation models, ranging from their capabilities (e.g., language, vision, robotic manipulation, reasoning, human interaction) and technical principles (e.g., model architectures, training procedures, data, systems, security, evaluation, theory) to their applications (e.g., law, healthcare, education) and societal impact (e.g., inequity, misuse, economic and environmental impact, legal and ethical considerations).

Though foundation models are based on standard deep learning and transfer learning, their scale results in new emergent capabilities, and their effectiveness across so many tasks incentivizes homogenization.

Fig. 2. A foundation model can centralize the information from all the data from various modalities. This one model can then be adapted to a wide range of downstream tasks.

GPT-4 (март 2023): тоже фундаментальная модель

GPT-4 visual input example, Solving École Polytechnique's Physics Exam:

User Answer question I.1.a. Think step-by-step.

I. Principe de la détection de rayonnement avec un bolomètre

Comme illustré sur la figure 1 un bolomètre est constitué d'un absorbeur qui reçoit le rayonnement que l'on désire détecter. Sa température T , supposée uniforme, est mesurée à l'aide d'un thermomètre incorporé, constitué d'un matériau conducteur dont la résistance $R(T)$ varie avec la température T ; cette variation est caractérisée par le coefficient $\alpha = \frac{1}{R} \frac{dR}{dT}$. L'ensemble possède la capacité thermique C_{th} .

Un barreau, conducteur thermique, homogène, de longueur L , de section S et de conductivité thermique λ et sans échanges thermiques latéraux, relie le bolomètre à un thermostat de température T_b fixe.

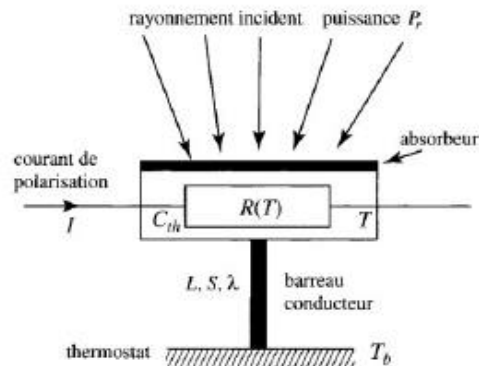


Figure 1 - Schéma d'un bolomètre.

I.1. Dans cette première partie, on étudie le comportement du bolomètre en régime stationnaire, à la température de fonctionnement $T = T_b$.

I.1.a) Exprimer la température $T(x)$ en tout point du barreau conducteur repéré par son abscisse x ; on prendra $T(0) = T_0$ et $T(L) = T_b$.

Так GPT и проходит, например, тесты по физике, где нужно анализировать схемы, формулы и текст одновременно

GPT-4 To answer question I.1.a, we need to express the temperature $T(x)$ in every point of the conductive bar, which is characterized by the abscissa x . We are given that $T(0) = T_0$ and $T(L) = T_b$.

For a stationary regime, the heat equation for the one-dimensional heat conduction in the bar is:

$$d^2T/dx^2 = 0$$

Integrating once, we get:

$$dT/dx = C_1$$

Integrating a second time:

$$T(x) = C_1x + C_2$$

To determine the constants C_1 and C_2 , we use the boundary conditions:

$$T(0) = T_0, \text{ which gives } C_2 = T_0$$

$$T(L) = T_b, \text{ which gives } C_1 = (T_b - T_0)/L$$

Finally, the expression for $T(x)$ is:

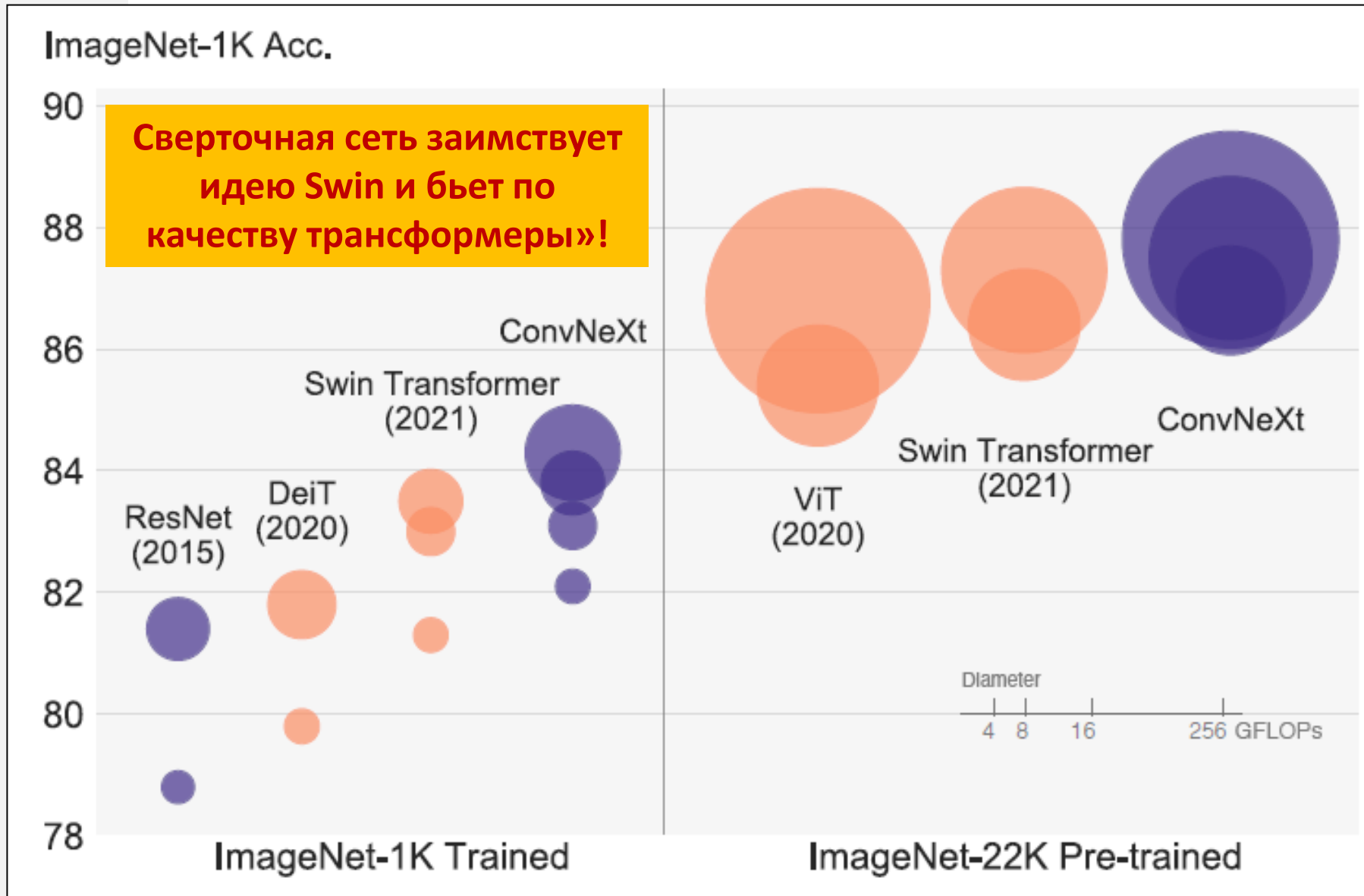
$$T(x) = (T_b - T_0) * (x/L) + T_0$$

This represents a linear temperature profile along the conductive bar, with a slope of $(T_b - T_0)/L$.

Теперь вход GPT можно подавать не только текст, но и изображения (правда, выход пока только текстовый)

ХИТ ИИ
2023!

Transformers vs. CNN: ничего еще не ясно! Например, ConvNeXt...



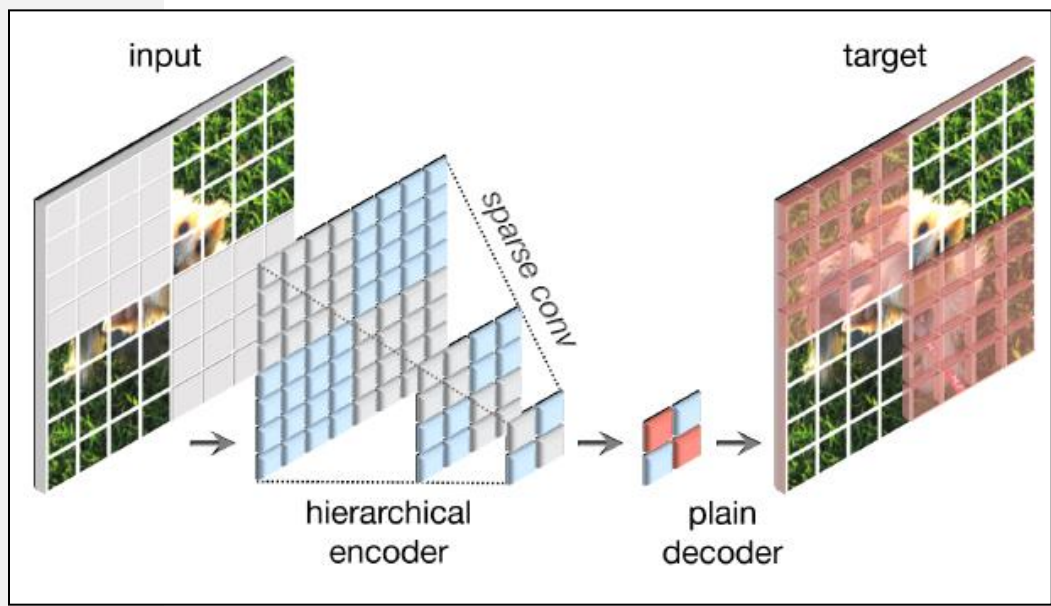
ImageNet-1K classification results for ConvNets and vision Transformers.

Each bubble's area is proportional to FLOPs of a variant in a model family. ImageNet-1K/22K models here take $224^2/384^2$ images respectively.

ResNet and ViT results were obtained with improved training procedures over the original papers.

We demonstrate that a standard ConvNet model can achieve the same level of scalability as hierarchical vision Transformers while being much simpler in design.

Transformers vs. CNN: ничего еще не ясно! ConvNeXt V2...(2023)



Fully Convolutional Masked Autoencoder

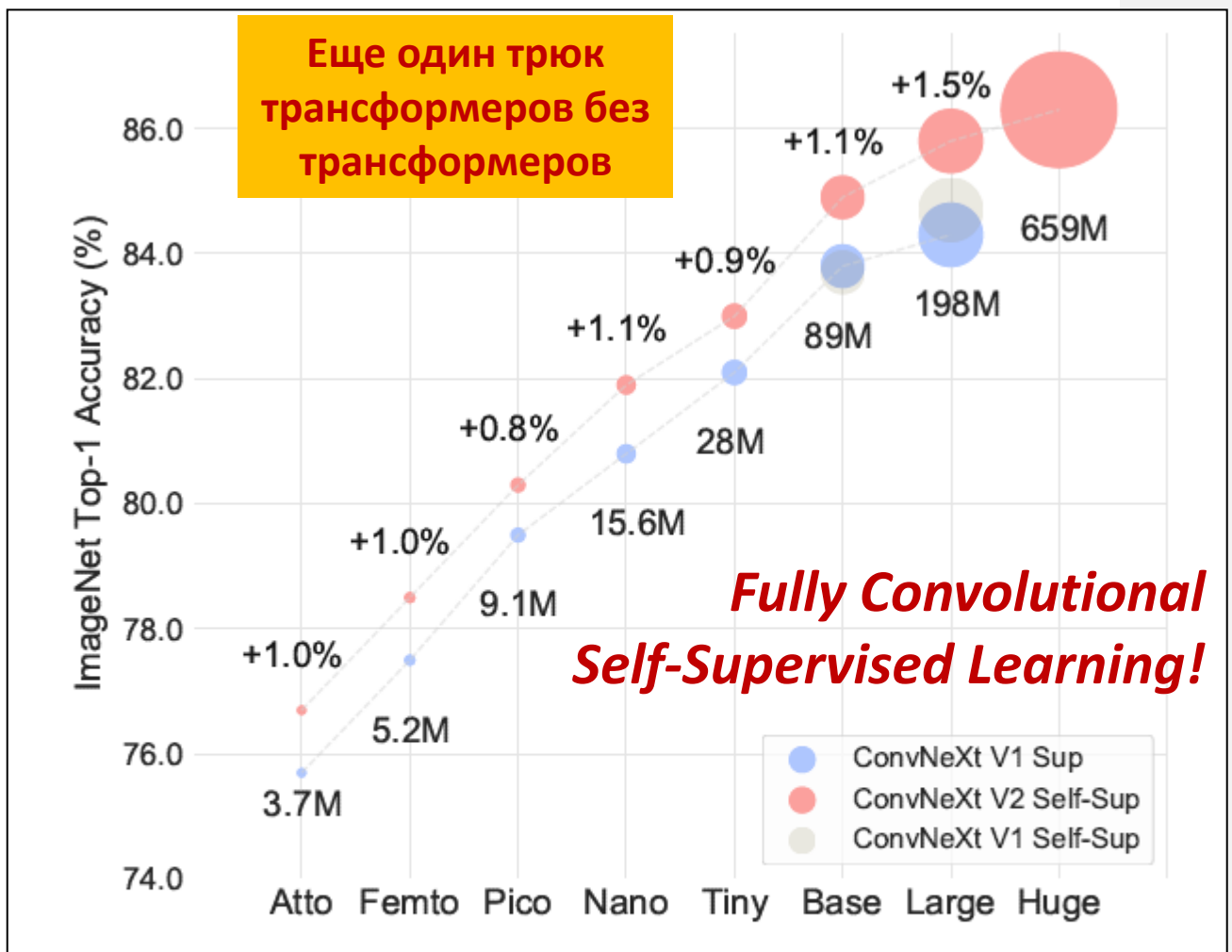
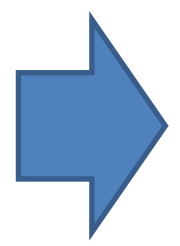
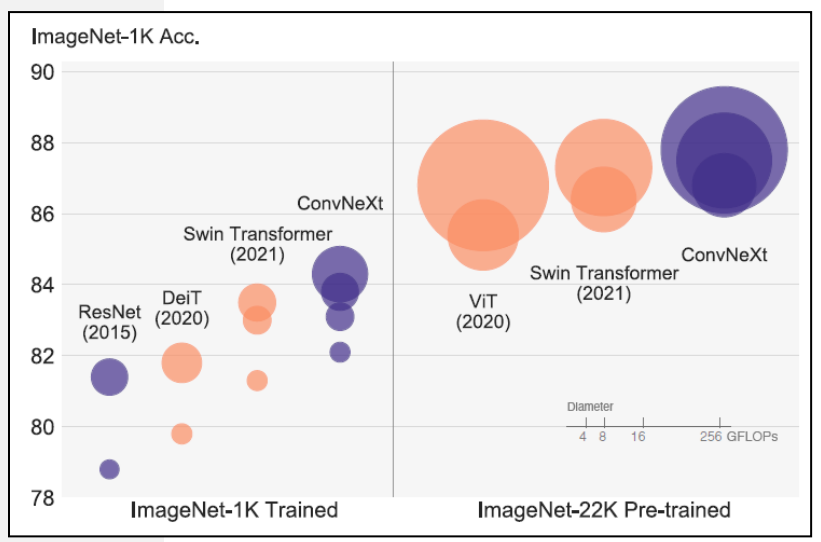
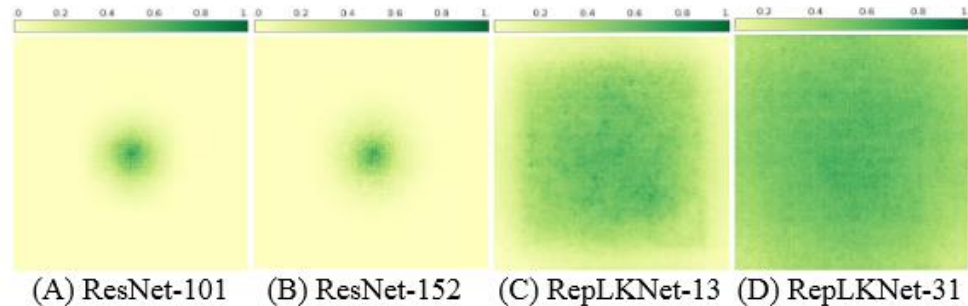


Figure 1. **ConvNeXt V2 model scaling.** The ConvNeXt V2 model, which has been pre-trained using our fully convolutional masked autoencoder framework, performs significantly better than the previous version across a wide range of model sizes.

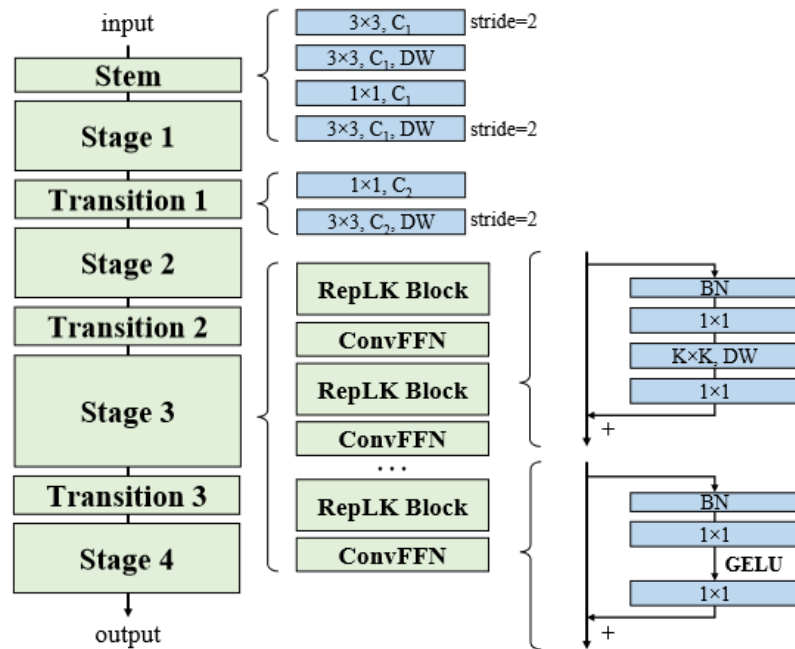
Transformers vs. CNN: ничего еще не ясно! Revisiting Large Kernels (RepLKNet)



*The **Effective Receptive Field (ERF)** of ResNet-101/152 and RepLKNet-13/31 respectively. A more widely distributed dark area indicates a larger ERF. More layers (e.g., from ResNet-101 to ResNet-152) help little in enlarging ERFs. Instead, our large kernel model **RepLKNet** effectively obtains large ERFs.

Briefly, we produce an aggregated contribution score matrix A (1024×1024), where each entry a ($0 \leq a \leq 1$) measures the contribution of the corresponding pixel on the input image to the central point of the feature map produced by the last layer.

***Scaling Up Your Kernels to 31x31: Revisiting Large Kernel Design in CNNs, Ding et al., 2022**



RepLKNet comprises Stem, Stages and Transitions. Except for depth-wise (DW) large kernel, the other components include DW 3×3 , dense 1×1 conv, and batch normalization (BN). Note that every conv layer has a following BN, which are not depicted. Such conv-BN sequences use ReLU as the activation function, except those before the shortcut-addition and those preceding GELU.

Model	Input resolution	Top-1 acc	Params (M)	FLOPs (G)	Throughput examples/s
RepLKNet-31B	224×224	83.5	79	15.3	295.5
Swin-B	224×224	83.5	88	15.4	226.2
RepLKNet-31B	384×384	84.8	79	45.1	97.0
Swin-B	384×384	84.5	88	47.0	67.9
RepLKNet-31B ‡	224×224	85.2	-	-	-
Swin-B ‡	224×224	85.2	-	-	-
RepLKNet-31B ‡	384×384	86.0	-	-	-
Swin-B ‡	384×384	86.4	-	-	-
RepLKNet-31L ‡	384×384	86.6	172	96.0	50.2
Swin-L ‡	384×384	87.3	197	103.9	36.2
RepLKNet-XL ◊	320×320	87.8	335	128.7	39.1

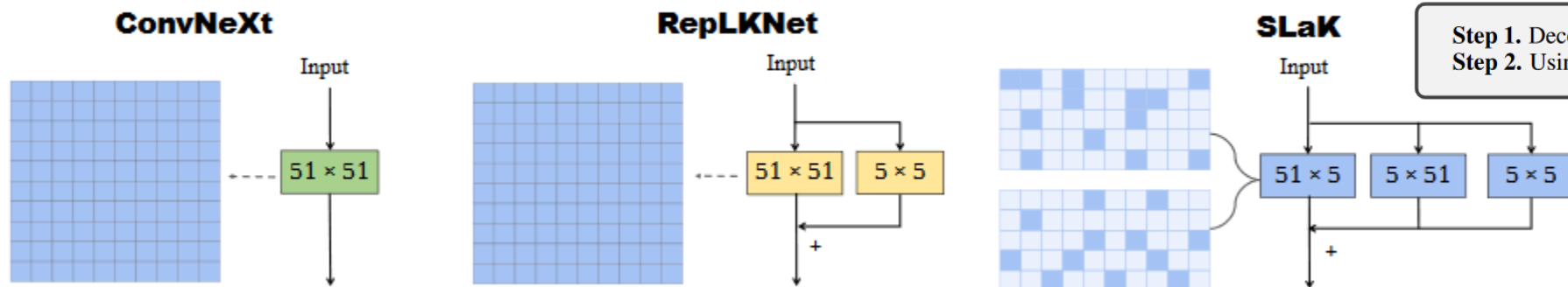
ImageNet results.

The throughput is tested with FP32 and a batch size of 64 on 2080Ti. ‡ indicates ImageNet-22K pretrain-ing.

◊ indicates pretrained with extra data.

Свертка с большими ядрами может эффективно моделировать свойства трансформеров!

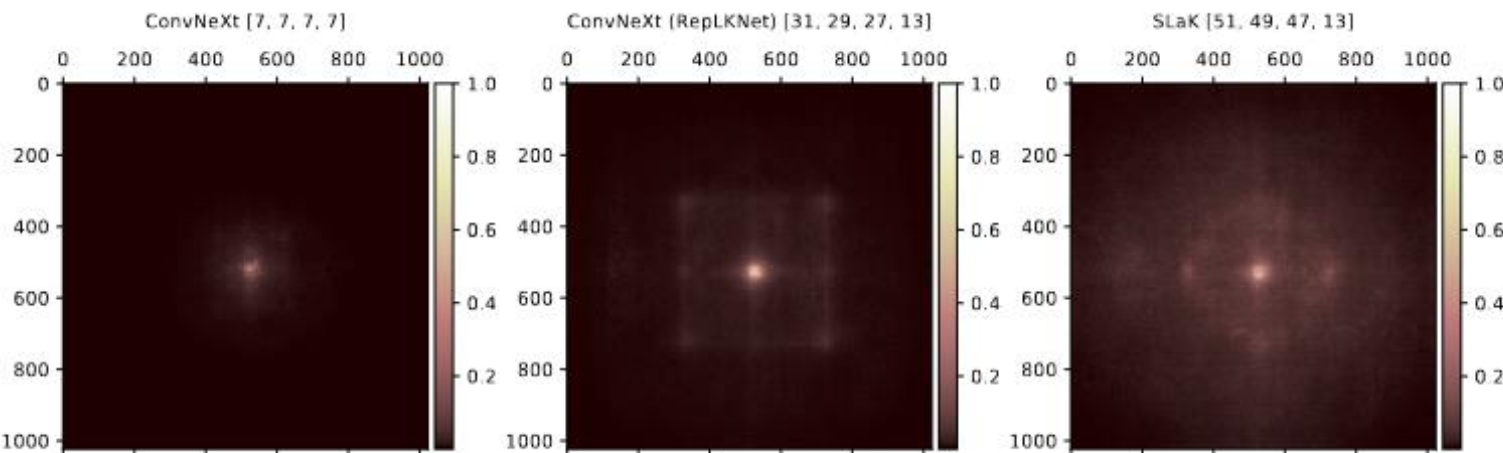
Transformers vs. CNN: ничего еще не ясно! Sparse Large Kernel Network (SLaK)



Разреженные ядра свертки могут быть еще больше и еще эффективнее

Large depth-wise kernel (e.g., 51×51) paradigms of ConvNeXt, RepLKNet, and SLaK. Dark blue squares refer to the dense weights in convolutional kernels. Light blue squares refer to the sparse weights in convolutional kernels.

Model	Image Size	#Param.	FLOPs	Top-1 Acc
ResNet-50 (He et al., 2016)	224×224	26M	4.1G	76.5
ResNeXt-50-32×4d (Xie et al., 2017)	224×224	25M	4.3G	77.6
ResMLP-24 (Touvron et al., 2021a)	224×224	30M	6.0G	79.4
DeiT-S (Touvron et al., 2021b)	224×224	22M	4.6G	79.8
Swin-T (Liu et al., 2021e)	224×224	28M	4.5G	81.3
TNT-S (Han et al., 2021a)	224×224	24M	5.2G	81.3
T2T-ViT _t -14 (Yuan et al., 2021a)	224×224	22M	6.1G	81.7
ConvNeXt-T (Liu et al., 2022b)	224×224	29M	4.5G	82.1
SLaK-T	224×224	30M/50M	5.0G/8.7G	82.5
Mixer-B/16 (Tolstikhin et al., 2021)	224×224	59M	11.6G	76.4
ResNet-101 (He et al., 2016)	224×224	45M	7.9G	77.4
ResNeXt101-32x4d (Xie et al., 2017)	224×224	44M	8.0G	78.8
PVT-Large (Wang et al., 2021b)	224×224	61M	9.8G	81.7
T2T-ViT _t -19 (Yuan et al., 2021a)	224×224	39M	9.8G	82.4
Swin-S (Liu et al., 2021e)	224×224	50M	8.7G	83.0
ConvNeXt-S (Liu et al., 2022b)	224×224	50M	8.7G	83.1
SLaK-S	224×224	55M/91M	9.8G/16.7G	83.8
DeiT-Base/16 (Touvron et al., 2021b)	224×224	87M	17.6G	81.8
RepLKNet-31B (Ding et al., 2022)	224×224	79M	15.3G	83.5
Swin-B (Liu et al., 2021e)	224×224	88M	15.4G	83.5
ConvNeXt-B (Liu et al., 2022b)	224×224	89M	15.4G	83.8
SLaK-B	224×224	95M/158M	17.1G/28.5G	84.0
ViT-Base/16 (Dosovitskiy et al., 2021)	384×384	87M	55.4G	77.9
DeiT-B/16 (Touvron et al., 2021b)	384×384	86M	55.4G	83.1
Swin-B (Liu et al., 2021e)	384×384	88M	47.1G	84.5
RepLKNet-31B (Ding et al., 2022)	384×384	79M	45.1G	84.8
ConvNeXt-B (Liu et al., 2022b)	384×384	89M	45.0G	85.1
SLaK-B	384×384	95M/158M	50.3G/83.8G	85.5



Effective receptive field (ERF) of models with various kernel sizes. SLaK is not only able to capture long-range dependence but also the local context features

***MORE CONVNETS IN THE 2020S: SCALING UP KERNELS BEYOND 51×51 USING SPARSITY, Liu et al., 2023**

Classification accuracy on ImageNet-1K. For SLaK models both theoretical, sparsity-aware numbers parameter & FLOPs (in black), and measured = no sparsity-aware acceleration (in blue).

Transformers vs. CNN: ничего еще не ясно! Math CNN Design (DeepMAD)

Theorem 1. The normalized Gaussian entropy upper bound of the MLP $f(\cdot)$ is

$$H_f = w_{L+1} \sum_{i=1}^L \log(w_i). \quad (1)$$

4.3. Final DeepMAD Formula

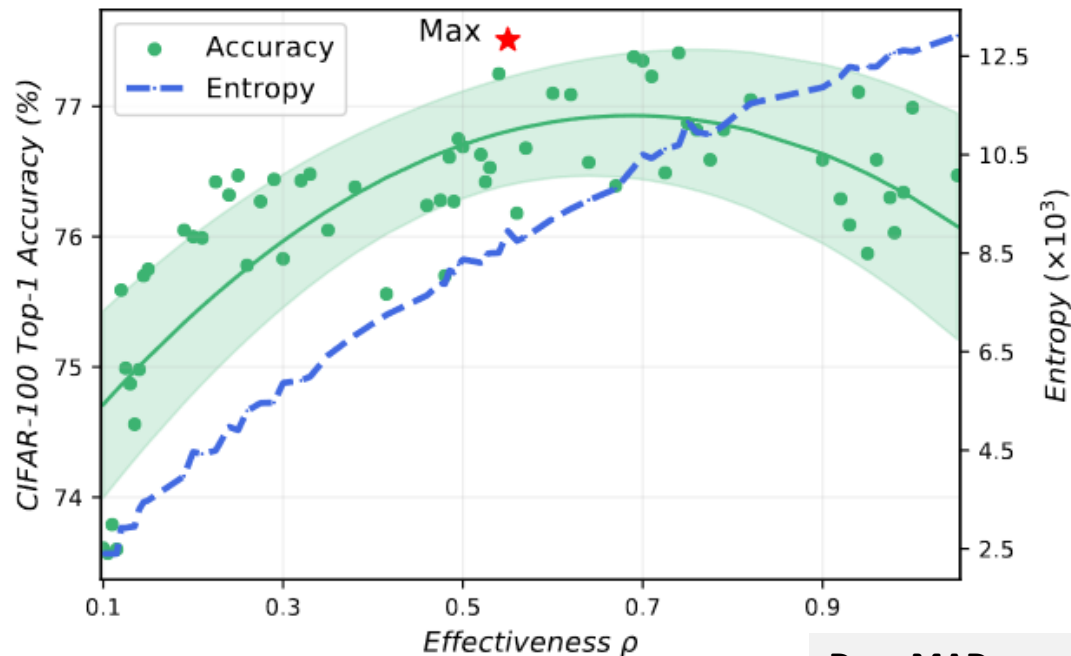
We gather everything together and present the final mathematical programming problem for DeepMAD. Suppose that we aim to design an L -layer CNN model $f(\cdot)$ with M stages. The entropy of the i -th stage is denoted as H_i defined in Eq. (4). Within each stage, all blocks use the same structural parameters (width, kernel size, etc.). The width of each CNN layer is defined by $w_i = c_i k_i^2 / g_i$. The depth of each stage is denoted as L_i for $i = 1, 2, \dots, M$. We propose to optimize $\{w_i, L_i\}$ via the following mathematical programming (MP) problem:

$$\begin{aligned} \max_{w_i, L_i} \quad & \sum_{i=1}^M \alpha_i H_i - \beta Q, \\ \text{s.t.} \quad & L \cdot \left(\prod_{i=1}^L w_i \right)^{-1/L} \leq \rho_0, \\ & \text{FLOPs}[f(\cdot)] \leq \text{budget}, \\ & \text{Params}[f(\cdot)] \leq \text{budget}, \\ & Q \triangleq \exp[\text{Var}(L_1, L_2, \dots, L_M)], \\ & w_1 \leq w_2 \leq \dots \leq w_L. \end{aligned} \quad (5)$$

Zero-Shot Auto-ML (DeepMAD):

Нужно оптимизировать соотношение глубины и ширины архитектурных блоков CNN, максимизируя энтропию (сложность) при ограничении эффективности (отношения глубины к средней ширине блока)

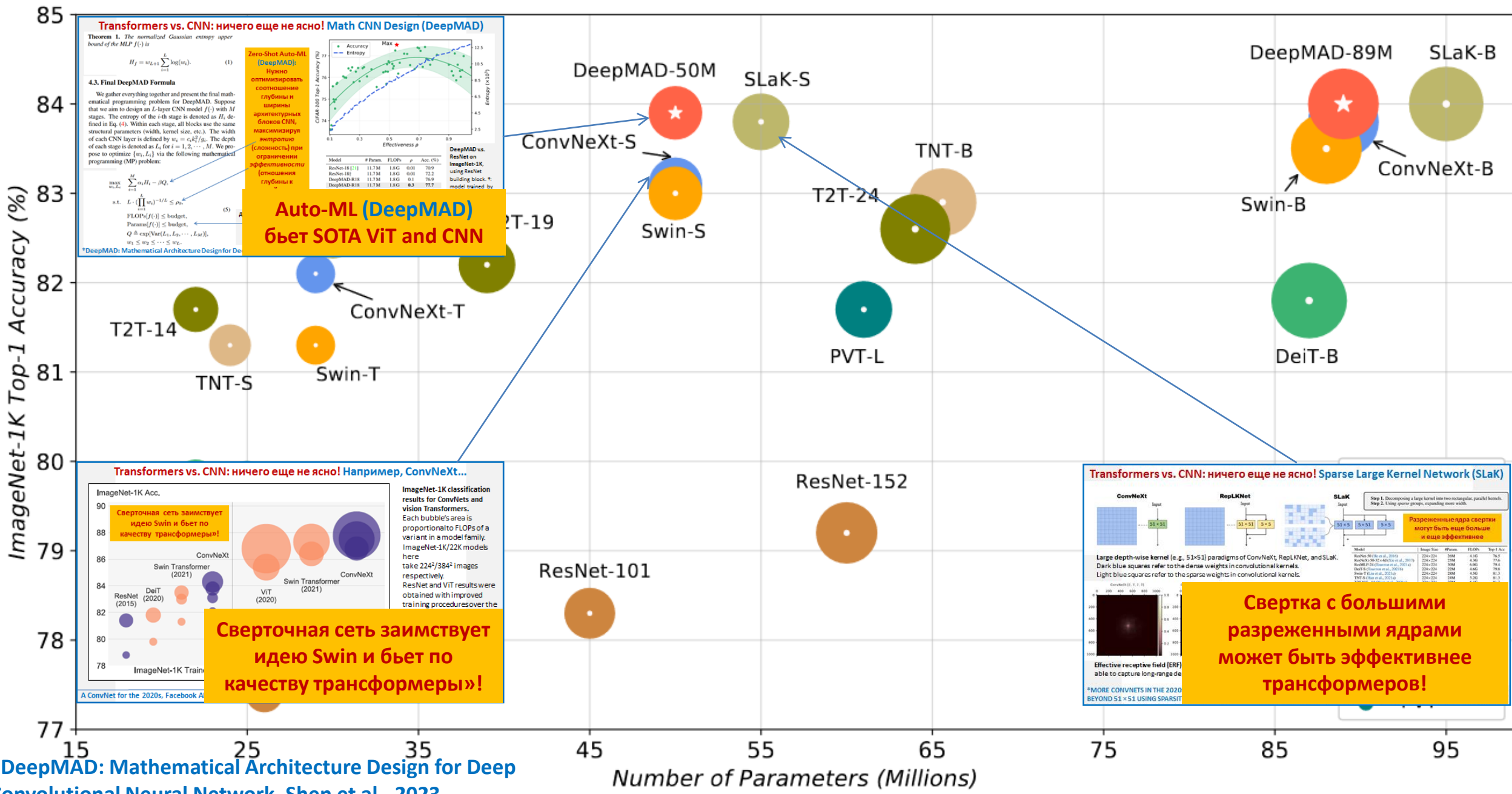
Даже с учетом дополнительных ограничений на число вычислений и параметров задача решается линейным программированием!



Model	# Param.	FLOPs	ρ	Acc. (%)
ResNet-18 [21]	11.7 M	1.8 G	0.01	70.9
ResNet-18†	11.7 M	1.8 G	0.01	72.2
DeepMAD-R18	11.7 M	1.8 G	0.1	76.9
DeepMAD-R18	11.7 M	1.8 G	0.3	77.7
DeepMAD-R18	11.7 M	1.8 G	0.5	77.5
DeepMAD-R18	11.7 M	1.8 G	0.7	75.7
ResNet-34 [21]	21.8 M	3.6 G	0.02	74.4
ResNet-34†	21.8 M	3.6 G	0.02	75.6
DeepMAD-R34	21.8 M	3.6 G	0.3	79.7
ResNet-50 [21]	25.6 M	4.1 G	0.09	77.4
ResNet-50†	25.6 M	4.1 G	0.09	79.3
DeepMAD-R50	25.6 M	4.1 G	0.3	80.6

DeepMAD v.s. ResNet on ImageNet-1K, using ResNet building block. †: model trained by our pipeline. ρ is tuned for DeepMAD-R18. DeepMAD achieves consistent improvements to ResNet18/34/50 with the same Params and FLOPs.

Transformers vs. CNN: ничего еще не ясно!



*DeepMAD: Mathematical Architecture Design for Deep Convolutional Neural Network, Shen et al., 2023

Машинное обучение для управления Reinforcement Learning (обучение с подкреплением)

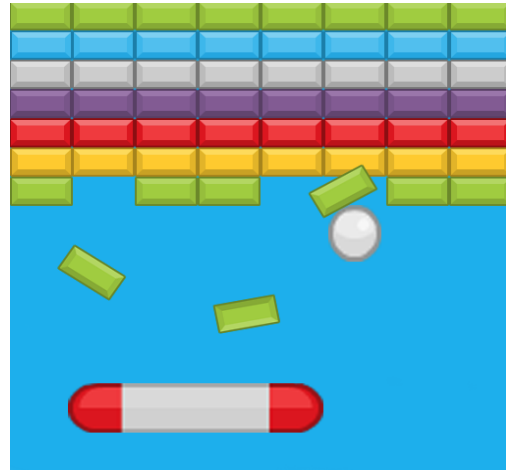
Advances in RL-2021-2023:

- ChatGPT (см.выше)
- Open-Ended Learning
- **GATO: A Generalist Agent**
- **Video PreTraining (VPT)**
- **DreamerV3: World Model**
- **GITM: LLM for RL**

Краткое напоминание

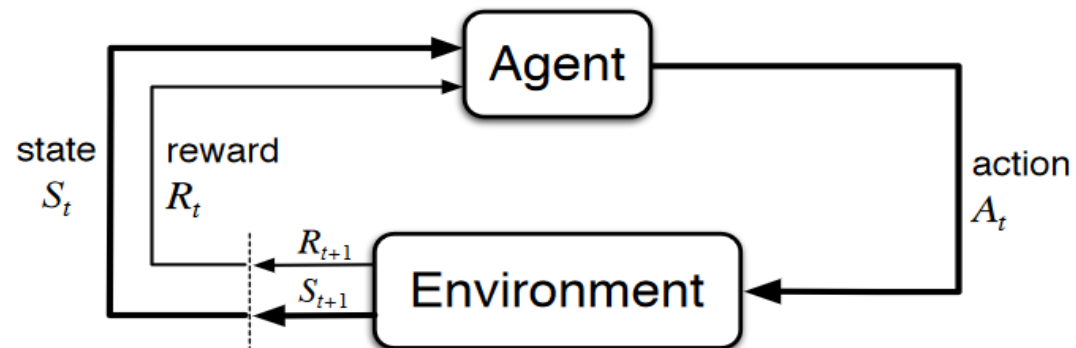
Главная идея: Когда нет возможности явной оптимизации дифференцируемой функции потерь, применяют обучение методом проб и ошибок за счет обратной связи от среды

Deep Reinforcement Learning: обучение с подкреплением



Агент (ОУ) действует в среде, и после каждого действия получает отклик в виде штрафа или вознаграждения

- **Reward $R(t)$** : score you earned at current step
Вознаграждение (после хода)
- **State S** : current screen
Состояние (что видим на экране)
- **Action i** : move your board left / right
Действие (управление/ход)
- **Policy $\pi(s)$** : How to choose your action
Стратегия (закон управления/хода)
- **Action value function $\hat{Q}(S, i)$** : your predicted future total rewards
Стоимость (выигрыш в будущем)



Задача RL в терминах теории управления: методом проб и ошибок найти закон управления, оптимизирующий функцию Беллмана

Как научить нейросеть играть в компьютерную игру

В 2019 много говорили о прорыве в RL для тактических игр и управления

AI Beats a Fighter Pilot in a Virtual Dogfight (2016)



AI ALPHA, built by Psibernetix, Inc. with Air Force Research Laboratory



Оперативно-тактическое и групповое управление



Показатель зрелости: игровые задачи

Сочетание технологий ГНС, RL и генетического отбора

2019+: Дорога открыта!



Автономное управление и оптимизация облика изделий

StarCraft II - полноценный тактический военный симулятор с упрощённой моделью ведения боя.

~~Прогноз 2018: ГНС смогут выиграть у людей лет через 5~~

25.01.2019: DeepMind AlphaStar со счетом 11:1 победила ведущих профессиональных игроков в StarCraft II

TOP AI 2019!

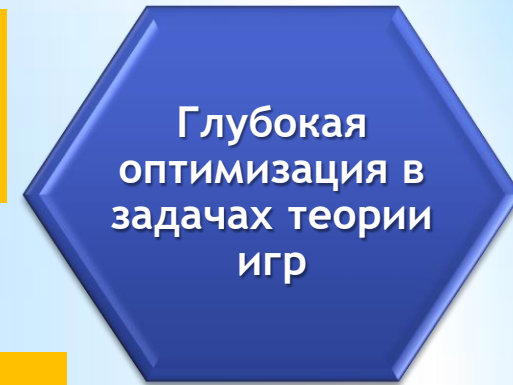


Оптимизация бизнес-процессов и производства

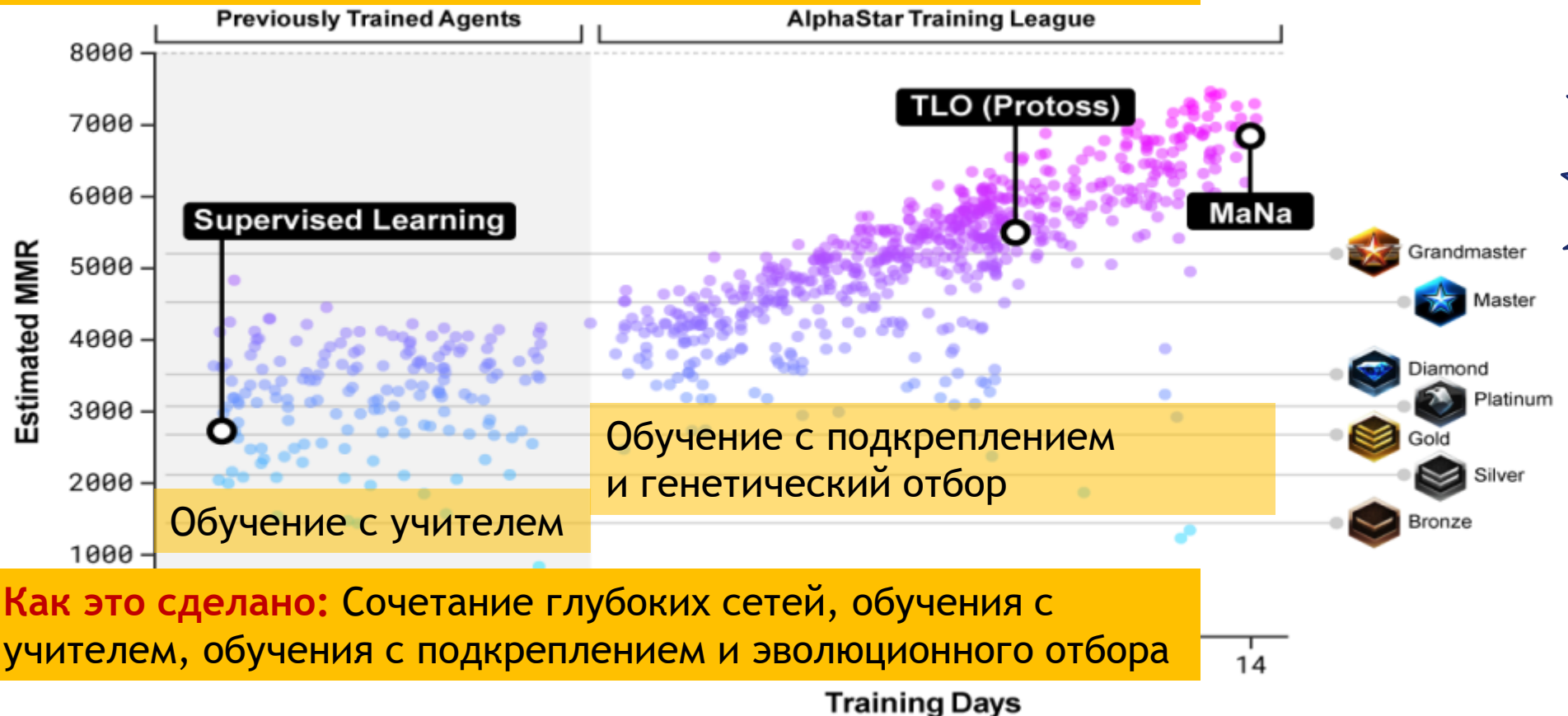
Технологии глубокого обучения в исследовании операций и управлении

25 января 2019 г. компания DeepMind опубликовала релиз: ГНС AlphaStar со счетом 11:1 победила ведущих профессиональных игроков в StarCraft II !!!

2019: ГНС строят модели тактических ситуаций и проводят анализ путей решения стратегических задач



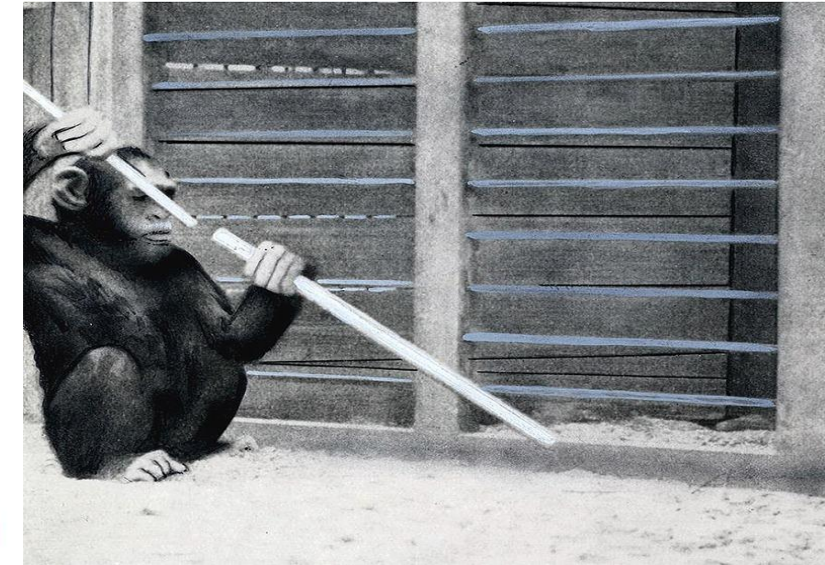
AlphaStar: Mastering the Real-Time Strategy Game StarCraft II



Как это сделано: Сочетание глубоких сетей, обучения с учителем, обучения с подкреплением и эволюционного отбора

Open-Ended Learning (Открытое обучение)

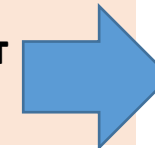
Можно ли научить ИИ решать заранее неизвестные задачи в совершенно новой обстановке?



Открытое обучение учит не задачам, а когнитивному поведению!

Полученные модели поведения (экспериментирование, использование инструментов, сотрудничество агентов) характерны для когнитивного поведения людей и животных и необходимы для самообучающихся роботов.

Гипотеза: если построить вселенную игровых задач и последовательно обучать ИИ-агентов играть в эти игры, то с каждой новой игрой они будут достигать лучших результатов в этой вселенной и за ее пределами



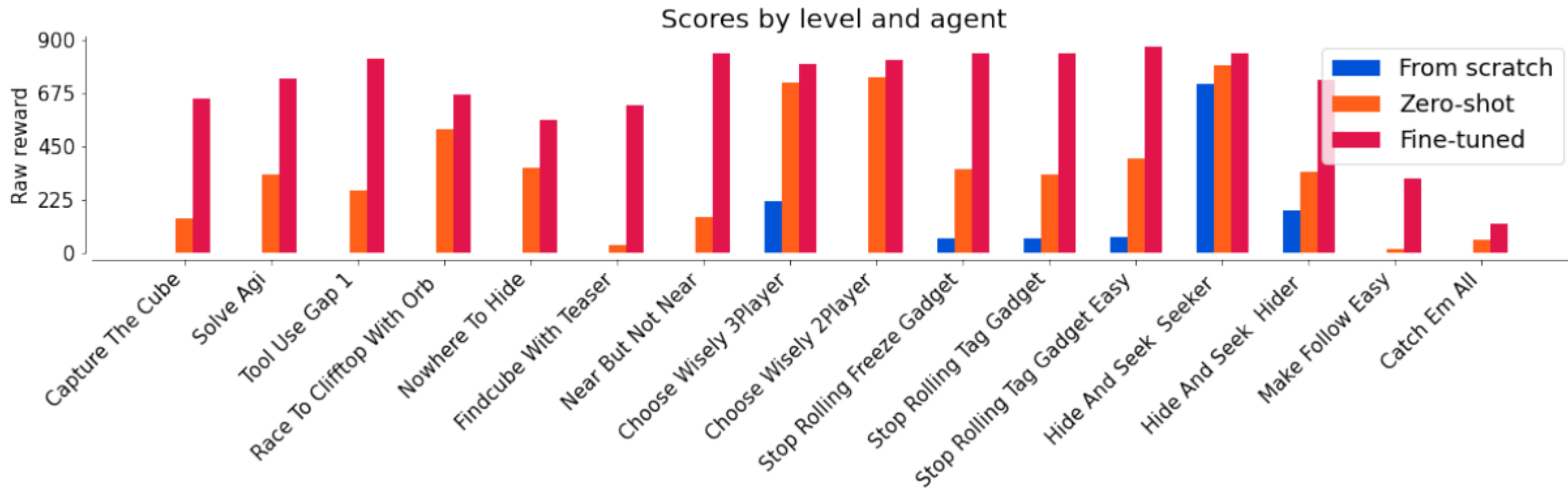
Open-Ended Learning Leads to Generally Capable Agents, DeepMind, 2021.

Open-Ended Learning

Открытое обучение учит не задачам, а когнитивному поведению!

Можно ли научить ИИ решать заранее неизвестные задачи в совершенно новой обстановке?

Но учиться (дообучаться) на конкретную задачу все равно нужно!



Comparison of three agents from different training regimes on a range of hand-authored levels. **Scratch:** An agent trained from scratch for 100 million steps. **Zero-shot:** the agent trained using our methodology and evaluated on these held out levels zero-shot. **Fine-tuned:** the same agent but trained for an additional 100 million steps on the level. 100 million steps is equivalent to 30 minutes of wall-clock time in our setup. **This rapid finetuning improves the agent score significantly compared to zero-shot, and in the majority of cases training from scratch does not achieve any reward.**

Close-Ended Learning vs. Open-Ended Learning

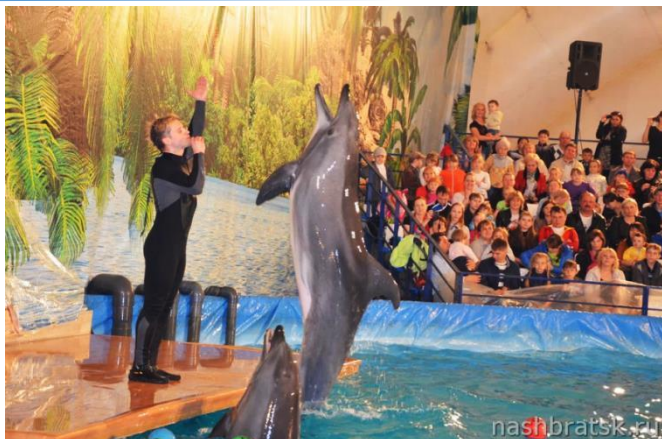
Произойдет ли смена парадигмы в обучении? Если да, то когда?

Close-Ended RL

Open-Ended RL

Можно ли научить ИИ решать заранее неизвестные задачи в совершенно новой обстановке?

Цель обучения:
обучение конкретным навыкам



Цель обучения:
обучение когнитивному поведению



Переход от схемы
1. обучение с учителем
2. обучение с подкреплением к трехэтапной, где первой фазой будет обучение с подкреплением когнитивному поведению



Процесс обучения

(I) Обучение когнитивному поведению

Обучение с подкреплением и отбор



Обучение с учителем

Обучение с подкреплением и отбор

(II) Изучение задачи

(III) Когнитивный поиск решения

Исходные данные и форма их представления

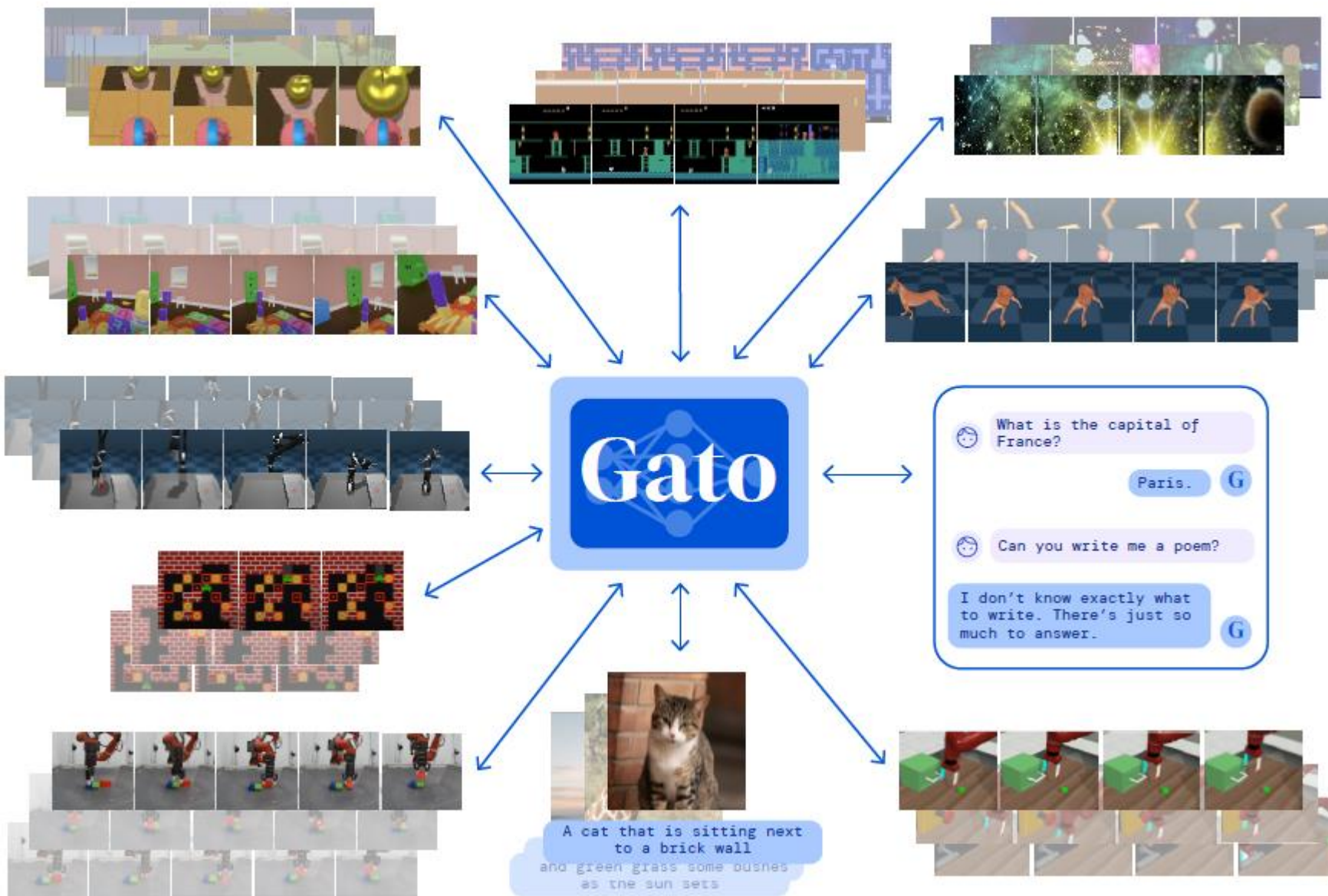
Задача представлена обучающей выборкой. Требуется набрать большой объем примеров действий людей.

Исходные данные и форма их представления

Задача представлена текстовым (формальным) описанием (zero-shot) и небольшой обучающей выборкой примеров (few-shot). Нужна база знаний предметной области (ИИ1 + ИИ2!).

GATO: универсальный агент

Трансформер для всех задач, включая управление



Inspired by progress in large-scale language modeling, we apply a similar approach towards building a **single generalist agent beyond the realm of text outputs**. The agent, which we refer to as Gato, works as a multi-modal, multi-task, multi-embodiment generalist policy. **The same network with the same weights can play Atari, caption images, chat, stack blocks with a real robot arm and much more**, deciding based on its context whether to output text, joint torques, button presses, or other tokens.

GATO: универсальный агент

Трансформер для всех задач, включая управление

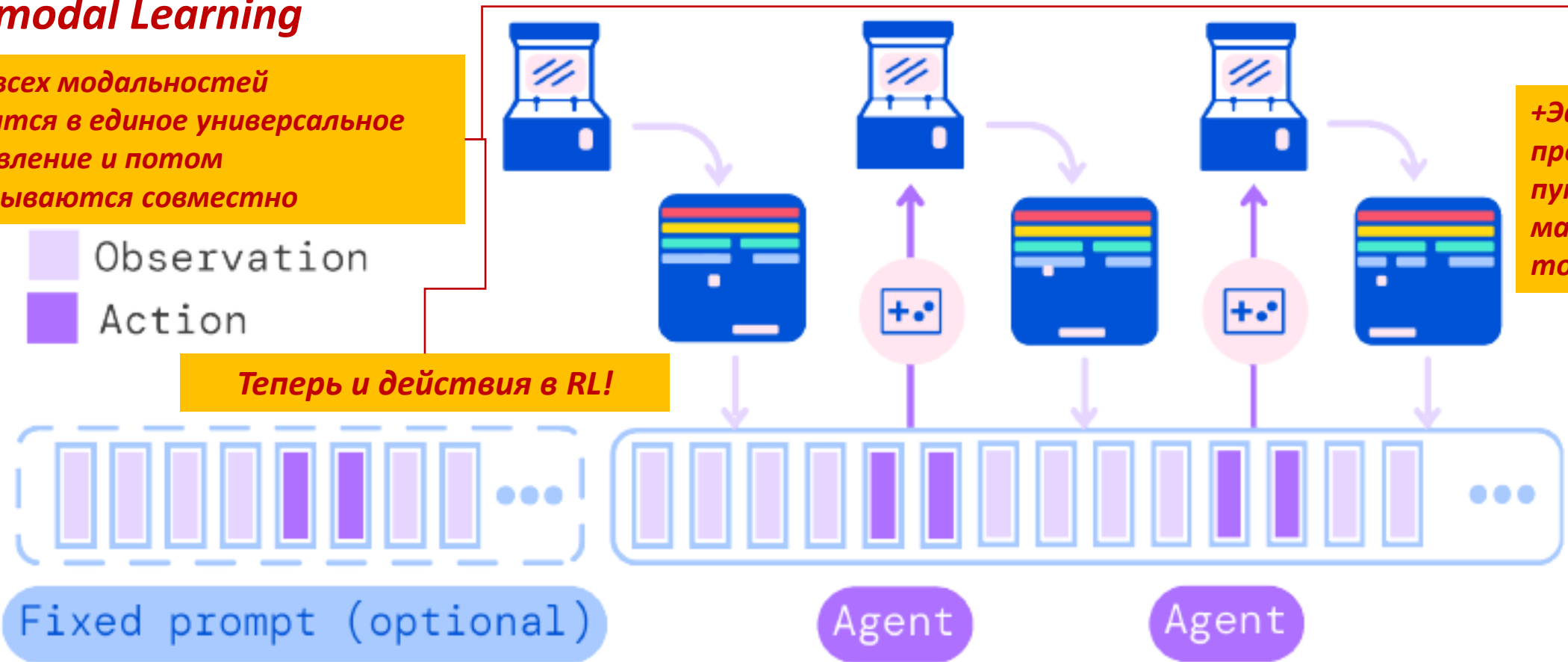
Multi-modal Learning

Данные всех модальностей переводятся в единое универсальное представление и потом обрабатываются совместно

Observation
Action

Теперь и действия в RL!

+Эффективное предобучение путем маскирования токенов



Running Gato as a control policy. Gato consumes a sequence of interleaved tokenized observations, separator tokens, and previously sampled actions to produce the next action in standard autoregressive manner. The new action is applied to the environment – a game console in this illustration, a new set of observations is obtained, and the process repeats.

Each batch mixes subsequences approximately uniformly over domains (e.g. Atari, MassiveWeb, etc.)!

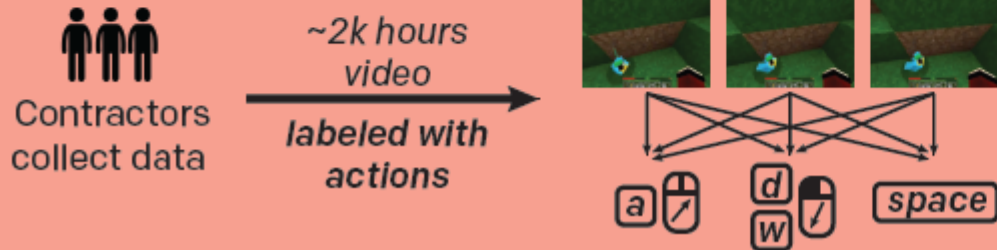
Video PreTraining (VPT)

Трансформер, который учится на неразмеченных примерах видео с игрой

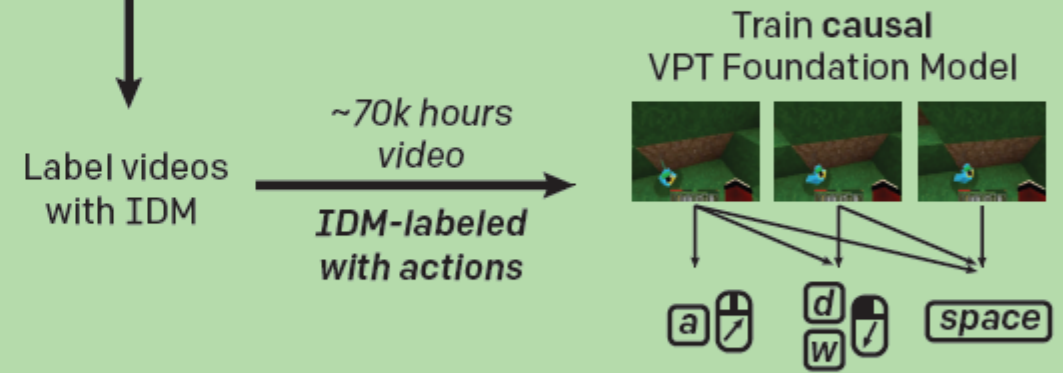
Collecting "Clean" Data



Training the Inverse Dynamics Model (IDM)



Training the VPT Foundation Model via Behavioral Cloning



Агент получил бриллиант в Minecraft

Для многих областей, требующих выполнения сложных последовательностей действий (робототехника, видеоигры,...), общедоступные данные не содержат меток, необходимых для обучения решению задач. Здесь предложена парадигма предварительного обучения по данным из Интернета на последовательное принятие решений с помощью имитации обучения, в ходе которой агенты учатся действовать, просматривая онлайн-видео без маркировки.

В частности, показано, что с небольшим количеством размеченных данных можно обучить модель обратной динамики, достаточно точную, чтобы разметить огромный неразмеченный источник онлайн-данных (**онлайн-видео людей, играющих в Minecraft**), на основе которых затем можно выучить общую модель поведения (игры).

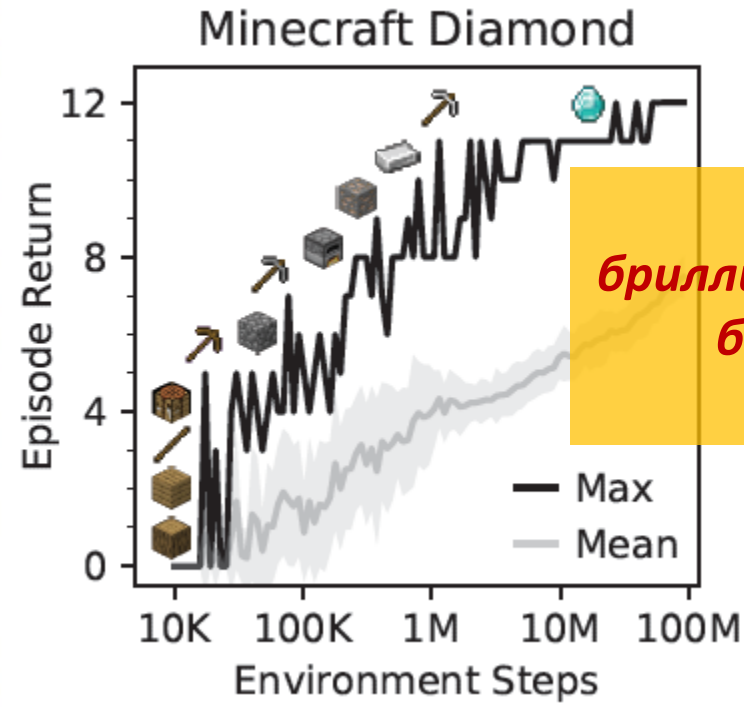
DreamerV3: World Model is all you need

Context Input Open Loop Prediction



Чтобы победить в игре, где вознаграждение отложено на много ходов, нужно уметь предсказывать будущее на много ходов!

Обучение с подкреплением на основе модели мира



Агент получил бриллиант в Minecraft без предобучения на видео!

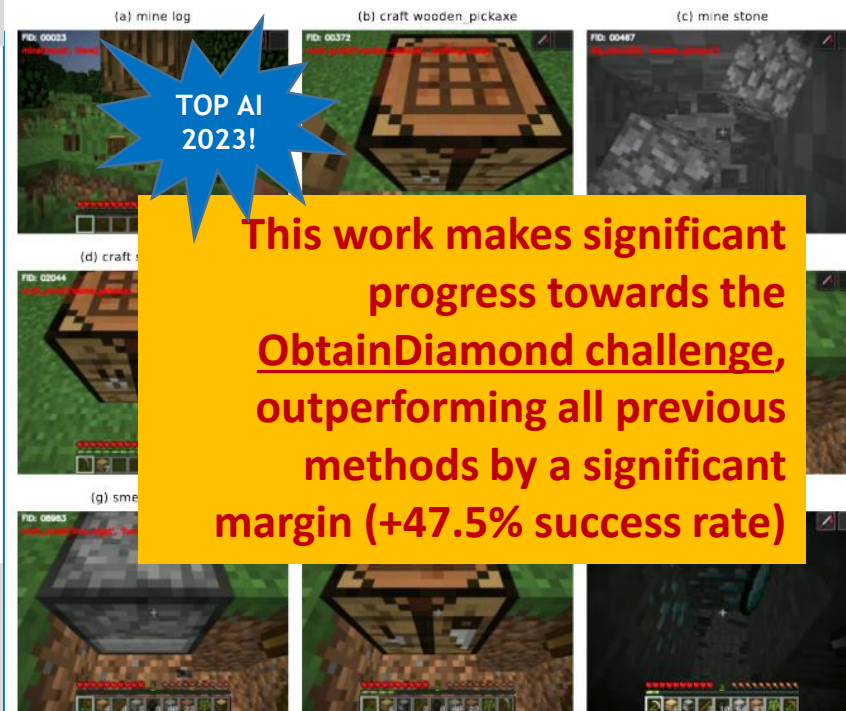
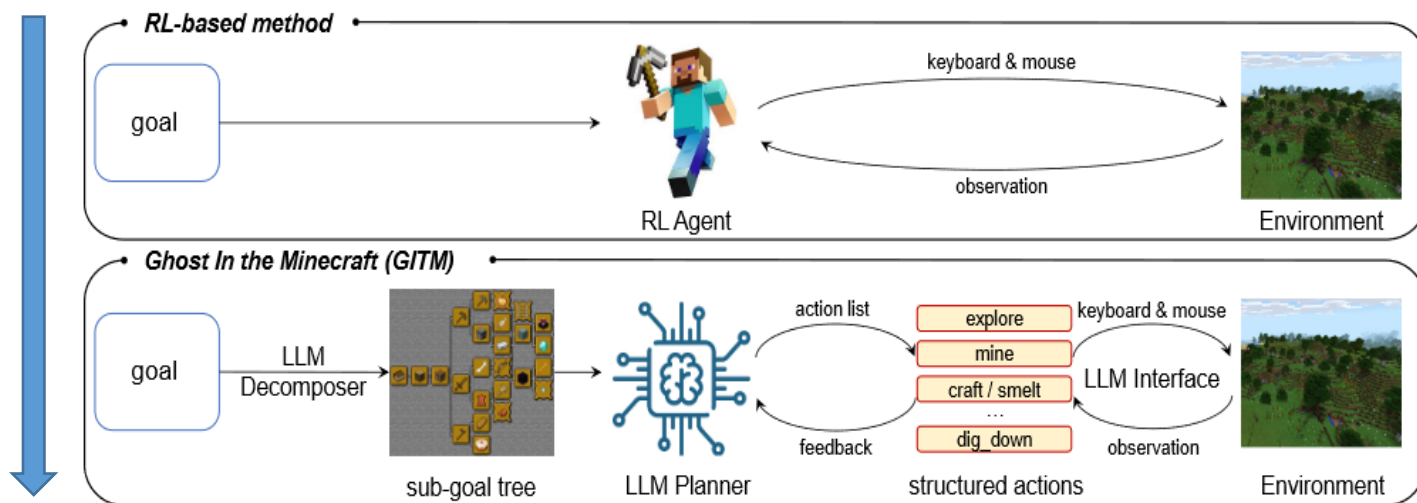
TOP AI 2022!

Multi-step predictions on Minecraft. The model receives the first 5 frames as context input and the predicts 45 steps into the future given the action sequence and without access to intermediate images

GITM: Large Language Model is all you need

Программирование на LLM для извлечения и применения знаний

RL как градиентный спуск: учим веса НС методом проб и ошибок



RL без градиентов: LLM, глядя в описание игры (Wiki Minecraft) выполняет автоматическое иерархическое планирование (построение дерева решений), которое в цикле проб и ошибок находит путь к заданной цели в игре

Стратег* разбивает задачу на части,
Тактик* планирует действия, Агент* играет, Наблюдатель* описывает опыт.
=> И так в цикле до достижения цели
*Все они – запросы к GPT-3.5!



(a) Shelter with Farmland



(b) Iron Golem



(c) Redstone Circuit



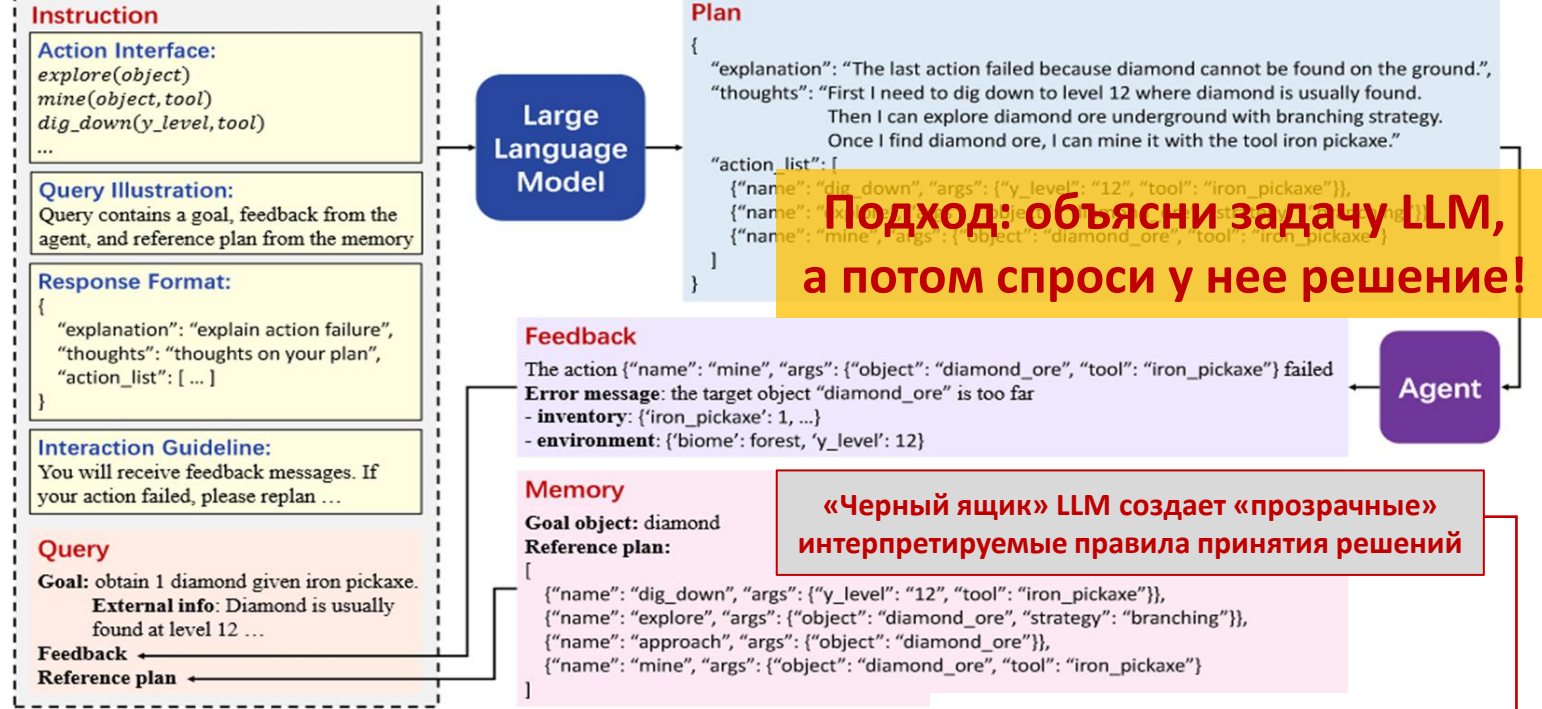
(d) Nether Portal

Ghost in the Minecraft: Generally Capable Agents for Open-World Environments via Large Language Models with Text-based Knowledge and Memory, 2023.

GITM: Large Language Model is all you need

Программирование на LLM для извлечения и применения знаний

RL без градиентов: не настройка весов, а извлечение знаний из проб и ошибок!



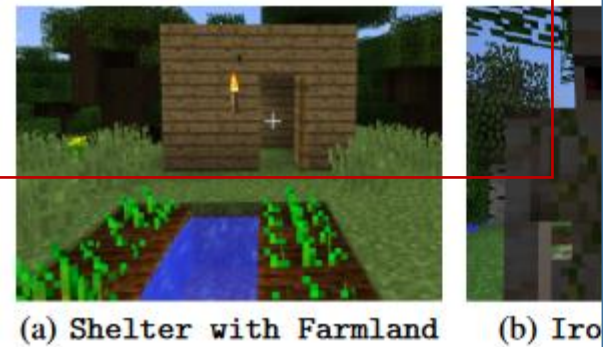
TOP AI 2023!

Пример новой парадигмы программирования в ИИ: Программирование на LLM для извлечения и применения знаний.

Частично напоминает программирование на Прологе. Отличие в том, что база знаний формируется автоматически в процессе обучения большой языковой модели. А вот составление запросов и использование ответов – дело «программистов на LLM».

Прогноз: в ближайшем будущем программисты на LLM во многом заменят программистов на Python. (Инженерия запросов)

Стратег* разбивает задачу на части, Тактик* планирует действия, Агент* играет, Наблюдатель* описывает опыт. => И так в цикле до достижения цели *Все они – запросы к GPT-3.5!



Инженерия запросов: Learning → Prompt Engineering

*Гибрид обучения и рассуждений,
путь к «прозрачному» объяснимому ИИ*

PE: Prompt Engineering is all you need!

Программирование на LLM для извлечения и применения знаний

Инженерия запросов: полезные ресурсы

The screenshot shows the homepage of <https://learnprompting.org/>. The page features a navigation menu on the left with categories like Basics, Basic Applications, Intermediate, Applied Prompting, Advanced Applications, Reliability, Image Prompting, Prompt Hacking, Tooling, Prompt Tuning, and Miscellaneous. The main content area has a 'Welcome' heading and a large illustration of an astronaut sitting on a rocky landscape. Below the illustration, there is a welcome message and a brief introduction to the course.

The screenshot shows the homepage of <https://www.promptingguide.ai/>. The page is in Russian and features a navigation menu on the left with categories like Введение, Настройки LLM, Основы промптинга, Элементы промпта, Общие советы по созданию промптов, Примеры промптов, and Техники. The main content area has a heading 'Руководство по промпт-инжинирингу' and a brief introduction to the field.

✓ **Промпт-инжиниринг - это новая дисциплина разработки и оптимизации запросов для эффективного использования языковых моделей (LM) в широком спектре приложений.**

✓ **Промпт-инжиниринг используется для повышения прозрачности и безопасности LLM, извлечения и добавления знаний, организации использования внешних инструментов и взаимодействия LLM.**

Basics: Introduction to prompt engineering and fundamental techniques

Basic Applications: Simple, practical applications of prompt engineering

Intermediate: Research-based PE techniques with moderate complexity

Applied Prompting: Comprehensive PE process walkthroughs contributed by community members

Advanced Applications: Powerful, and more complex applications of prompt engineering

Reliability: Enhancing the reliability of large language models (LLMs)

Image Prompting: Prompt engineering for text-to-image models, such as DALL-E and Stable Diffusion

Prompt Hacking: Hacking, but for prompt engineering

Tooling: A review of various prompt engineering tools and IDEs

Prompt Tuning: Refining prompts using gradient-based techniques

Miscellaneous: A collection of additional topics and techniques related to prompt engineering

Chain-of-Thought Prompting: проблема не только в обучении!

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

<input, chain of thought, output> triples

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅

Chain of thought:

1. allows models to **decompose multi-step problems into intermediate steps**,
2. provides an **interpretable window into the behavior of the model**, suggesting how it might have arrived at a particular answer and providing opportunities to **debug where the reasoning path went wrong!**

Спрашивайте LLM правильно: просите ответ + этапы рассуждений => точность ответа вырастает!

Много ошибок, почти все «на троечку»

Chain-of-thought prompting enables LLM to tackle complex arithmetic, commonsense, and symbolic reasoning tasks

Algorithm of Thoughts: от цепочек к сложным рассуждениям

Standard Prompting

Model Input

User: 8 6 4 4
 Assistant:
 Answer: $(4 + (8 - 6)) * 4 = 24$.
 User: 11 10 5 1

Model Output

Assistant:
 Answer: $(11 - 1) * (10 - 5) = 24$

Chain-of-Thought

Model Input

User: 8 6 4 4
 Assistant:
 $8 - 6 = 2$ (left: 4, 4, 2)
 $4 + 2 = 6$ (left: 6, 4)
 $6 * 4 = 24$
 Answer: $(4 + (8 - 6)) * 4 = 24$.
 User: 11 10 5 1

Model Output

Assistant:
 $11 - 10 = 1$ (left: 1, 1, 5)
 $1 * 1 = 1$ (left: 1, 5)
 $5 * 1 = 5$ (left: 5)
 Answer: $((11 - 10) * 1) * 5 - 1 = 24$.

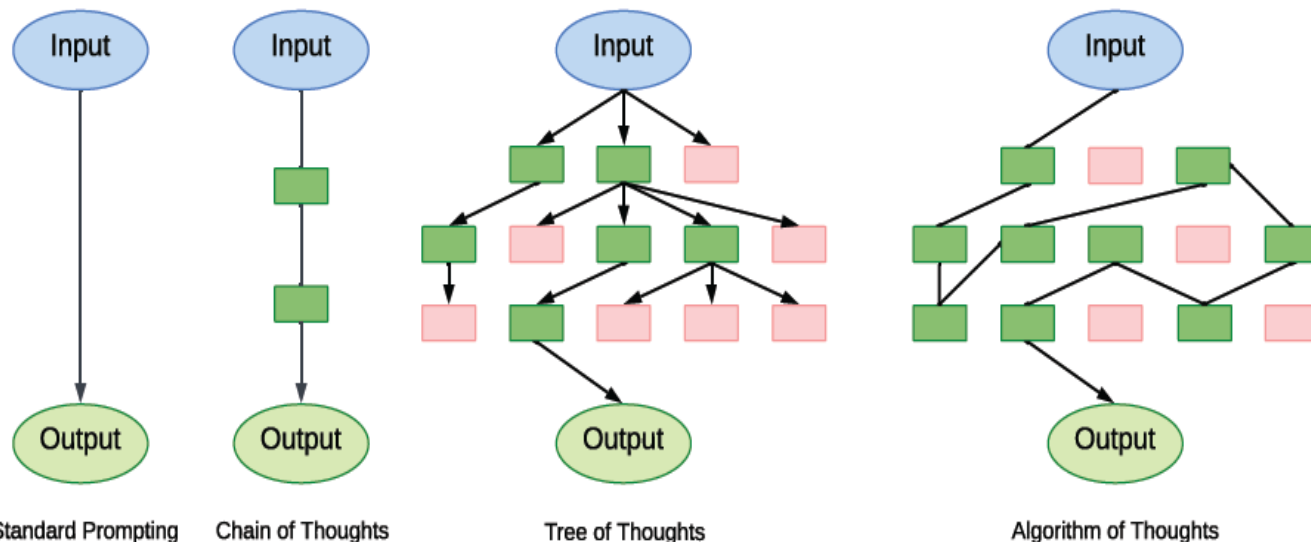
Algorithm of Thoughts

Model Input

User: 8 6 4 4
 Assistant:
 1. $4 - 4$: (left: 8, 6, 0)
 .
 .
 3. $8 - 6$: (left: 4, 4, 2)
 $- 4 + 4$: (left: 8, 2) 10, 6, 16, 4
 $- 4 - 4$: (left: 2, 0) 2, 2, 0, undefined
 $- 4 * 4$: (left: 16, 2) 18, 14, 32, 8
 $- 4 / 4$: (left: 2, 1) 3, 1, 2, 2
 $- 4 + 2$: (left: 6, 4) 10, 2, 24 = $6 * 4$ -> found it!
 Answer: $(4 + (8 - 6)) * 4 = 24$.
 User: 11 10 5 1

Model Output

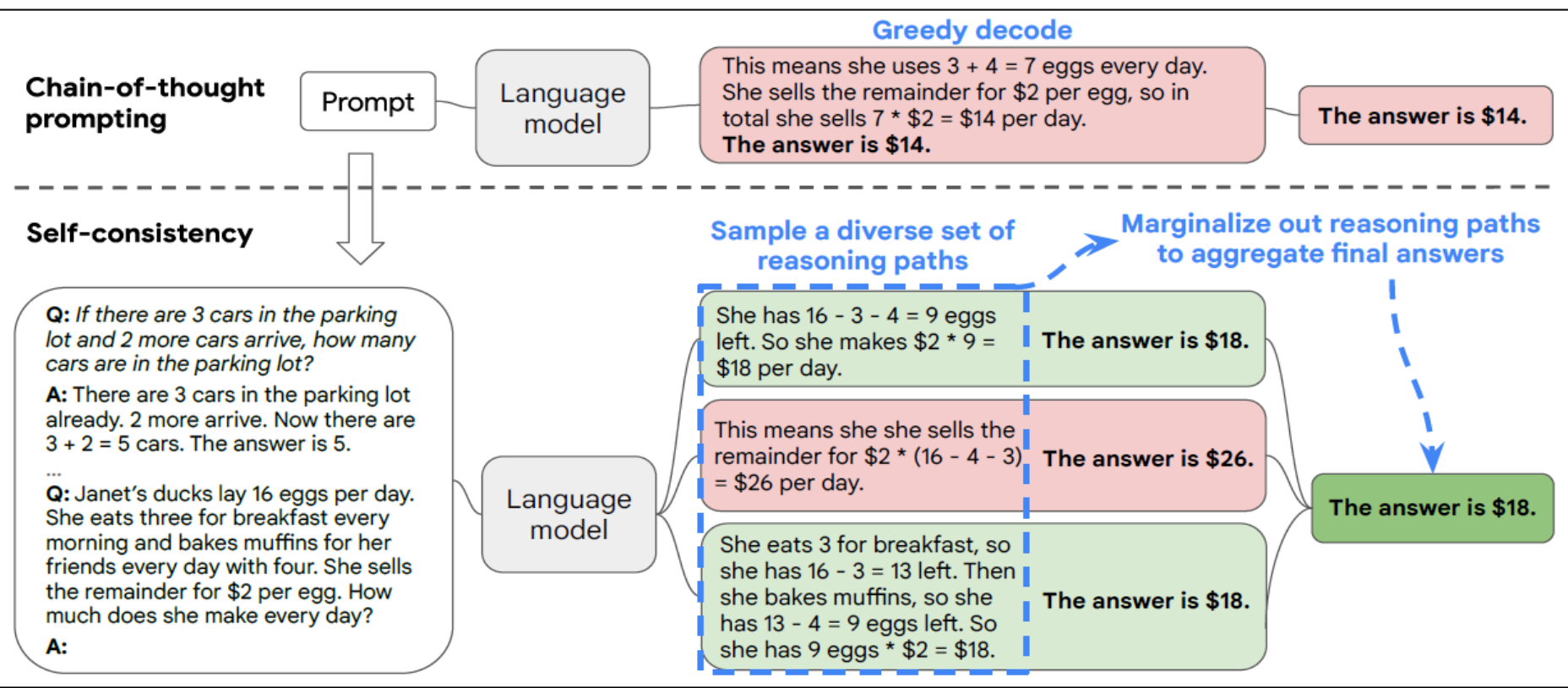
Assistant:
 1. $11 - 1$: (left: 10, 10, 5)
 .
 .
 5. $11 + 1$: (left: 12, 10, 5)
 $- 12 + 10$: (left: 22, 5) 27, 17, 110, 4.4
 $- 12 - 10$: (left: 5, 2) 7, 3, 10, 2.5
 $- 12 * 10$: (left: 120, 5) 24 = $120 / 5$ -> found it!
 Answer: $((11 + 1) * 10) / 5 = 24$.



Various strategies for tackling reasoning problems with LLMs. Each box signifies a distinct thought, functioning as a unified string of words that forms an incremental pathway to reasoning. Green boxes indicate ideas deemed promising by the LLM, while red boxes represent less promising concepts.

Comparison between standard prompting, CoT, and AoT in the game of 24. CoT sketches out the successive steps to the final solution. AoT's in-context example, distinct from CoT, integrates the search process, highlighted by markers '1',..., '3' as "first operations" guiding subtree exploration for the problem set '8 6 4 4'. For clarity, only a single in-context example is displayed, with a focus on the third subtree exploration. AoT produces prospective search steps (e.g., the subtree exploration '5. 11 + 1') and evaluates potential subsequent steps to either progress towards a solution or retrace to another viable subtree.

Self-Consistency: заставьте LLM изобретать разные пути решения



Чтобы найти разные решения, LLM нужно больше фантазии!

***Temperature - LLM parameter**, The lower the temperature, the more deterministic the results. Usually a lower temperature value is for tasks like fact-based QA to make LLM more factual.

For poem generation or other creative tasks, it might be beneficial to increase the temperature value.

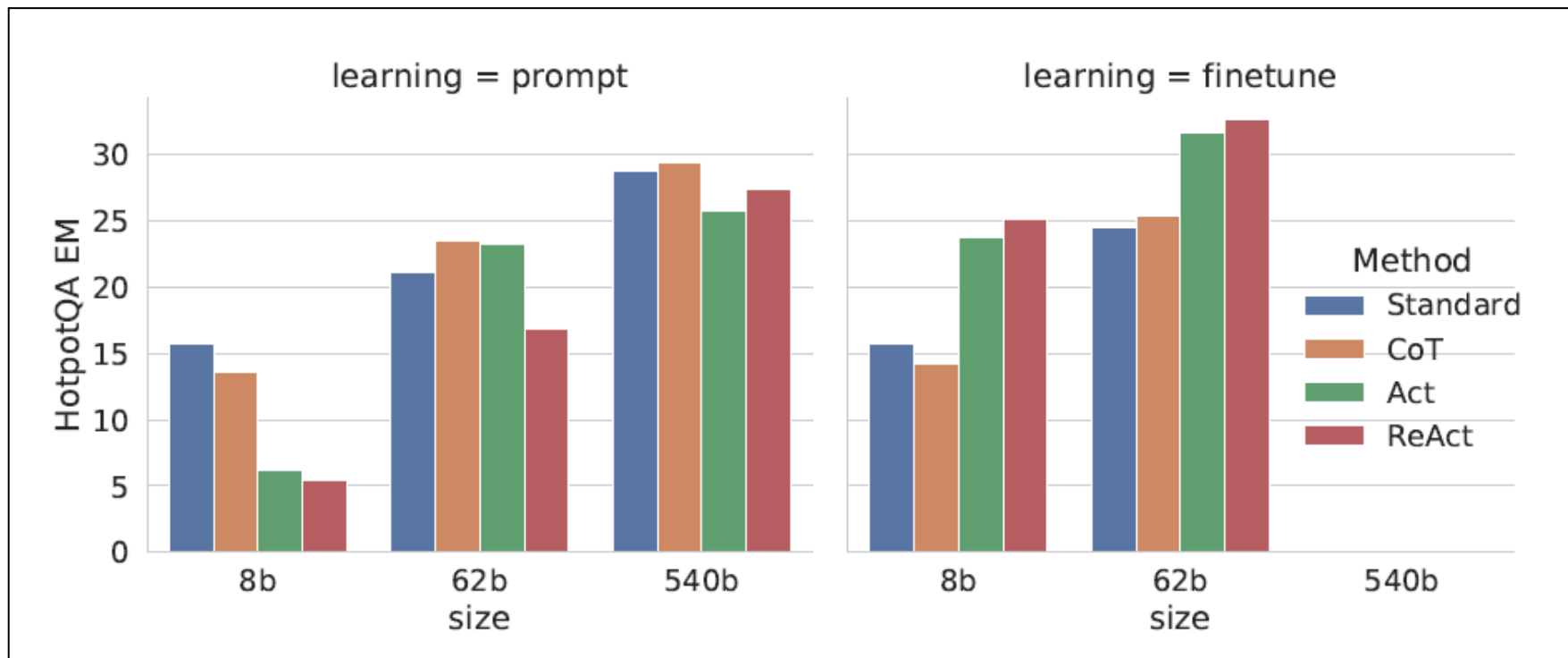
Self-Consistency acts like a “self-ensemble” on top of a single LLM. SC is between open-ended text generation and generation of fixed answer.

Спрашивайте LLM правильно: запросите разный ход решения и выберите совпадающий ответ!

Много ошибок, почти все «на троечку»

Self-consistency leverages the intuition that a complex reasoning problem typically admits multiple different ways of thinking leading to its unique correct answer.

Промптинг + дообучение: модели становятся меньше и работают быстрее за счет доведения интеллектуальных действий до автоматизма

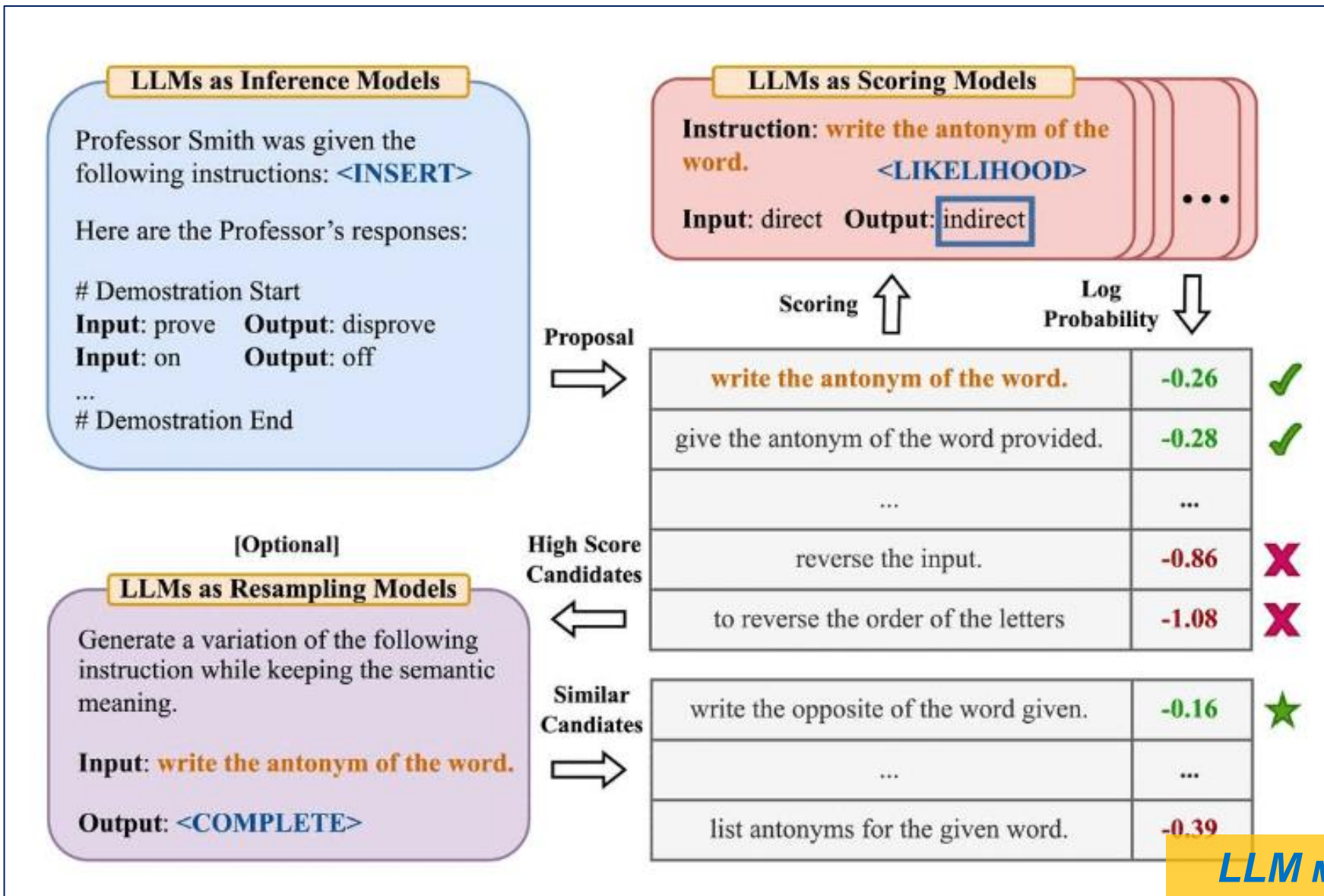


Как и в открытом обучении, все-таки промптинг и рассуждения **требуют** **файнтюнинга на конкретные виды деятельности** (игры), но он может быть очень небольшим (на **сотнях-тысячах** примеров). При этом результат достигается на **порядок меньшей моделью!**

ReAct performs best for fine-tuning. Figure shows the scaling effect of prompting/finetuning four methods (Standard, CoT, Act, ReAct) on HotpotQA. With PaLM-8/62B, prompting ReAct performs worst among four methods. However, when **finetuned with just 3,000 examples, ReAct becomes the best method, with PaLM-8B finetuned ReAct outperforms all PaLM-62B prompt methods, and PaLM-62B finetuned ReAct outperforms all 540B prompting methods.**

Дообучение LLM на примерах собственного успешного промптинга это **перевод навыков LLM с «сознательного» уровня на «бессознательный», закрепление действий, доведение до автоматизма**

APE: Automatic Prompt Engineer (APE)



Automatic Prompt Engineer (APE)

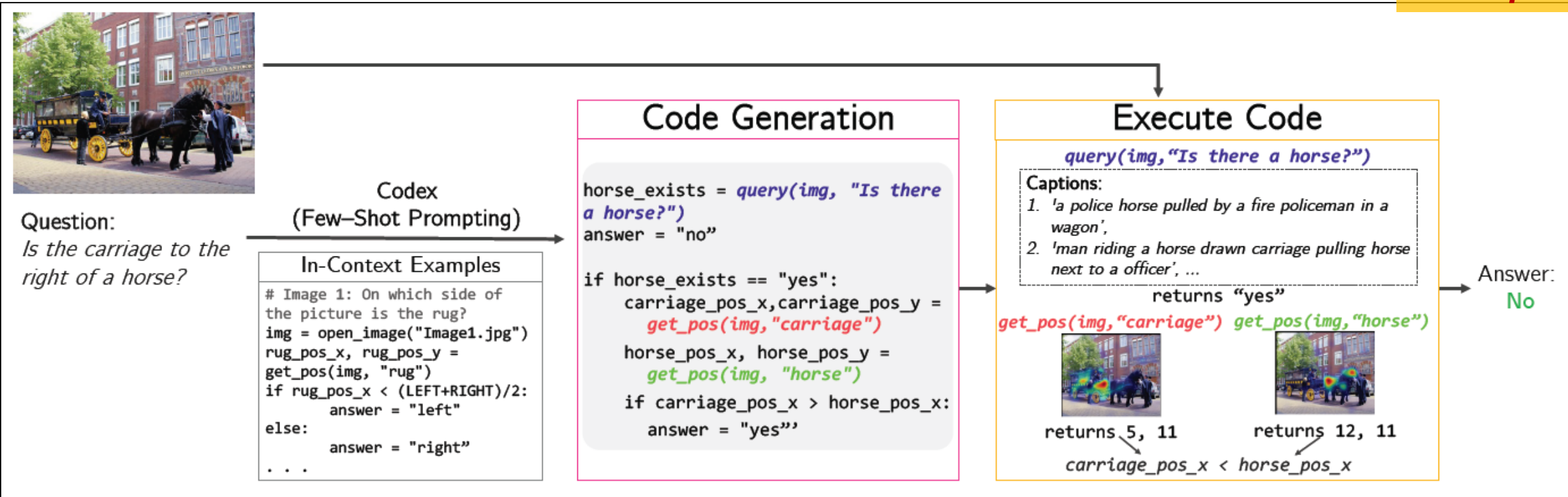
Zhou et al., (2022) propose automatic prompt engineer (APE) a framework for automatic instruction generation and selection.

The instruction generation problem is framed as natural language synthesis addressed as a black-box optimization problem using LLMs to generate and search over candidate solutions.

LLM могут не только отвечать на запросы, но и генерировать их не хуже человека!

CodeVQA: Visual Question Answering via Code Generation

Image Prompting



CodeVQA Overview. CodeVQA first prompts Codex with in-context examples that break down a given question into Python code. Using just the question, Codex generates an executable program that composes pre-defined visual modules using conditional logic, arithmetic, etc. The visual module, query answers a question by captioning the image and using an LM to answer based on the captions. get_pos retrieves the location of the object. Here, CodeVQA correctly identifies the question as a conjunction of a query and a spatial comparison and arrives at the right answer.

2020: Трансформеры сделали программирование ненужным для VQA. 2023: LLM и PE его вернули!

Visual Programming (VISPROG): Visual reasoning without training

Visual Programming

Prediction Visual Rationale

VISPROG

Program Interpreter

High-level Program

VISPROG

Program Generator

Input Image(s)

Natural Language Instruction

In-context instruction-program pairs

Compositional Visual Question Answering

IMAGE:



Question: Are there both ties and glasses in the picture?

Program:

```
BOX0=Loc(image=IMAGE, object='ties')
ANSWER0=Count(box=BOX0)
BOX1=Loc(image=IMAGE, object='glasses')
ANSWER1=Count(box=BOX1)
ANSWER2=Eval("'yes' if {ANSWER0} > 0 and {ANSWER1} > 0 else 'no'")
RESULT=ANSWER2
Prediction: no
```

Natural Language Visual Reasoning

LEFT:



RIGHT:



Statement: The left and right image contains a total of six people and two boats.

Program:

```
ANSWER0=Vqa(image=LEFT, question='How many people are in the image?')
ANSWER1=Vqa(image=RIGHT, question='How many people are in the image?')
ANSWER2=Vqa(image=LEFT, question='How many boats are in the image?')
ANSWER3=Vqa(image=RIGHT, question='How many boats are in the image?')
ANSWER4=Eval('{ANSWER0} + {ANSWER1} == 6 and {ANSWER2} + {ANSWER3} == 2')
RESULT=ANSWER4
Prediction: False
```

Factual Knowledge Object Tagging

IMAGE:



Prediction: IMAGE0



Instruction: Tag the 7 main characters on the TV show Big Bang Theory

Program:

```
OBJ0=FaceDet(image=IMAGE)
LIST0=List(query='main characters on the TV show Big Bang Theory', max=7)
OBJ1=Classify(image=IMAGE, object=OBJ0, categories=LIST0)
IMAGE0=Tag(image=IMAGE, object=OBJ1)
RESULT=IMAGE0
```

Natural Language Image Editing

IMAGE:



Prediction: IMAGE1



Instruction: Hide Daniel Craig with 8) and Sean Connery with ;)

Program:

```
OBJ0=FaceDet(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='Daniel Craig', category=None)
IMAGE0=Emoji(image=IMAGE, object=OBJ1, emoji='smiling_face_with_sunglasses')
OBJ2=Select(image=IMAGE, object=OBJ0, query='Sean Connery', category=None)
IMAGE1=Emoji(image=IMAGE0, object=OBJ2, emoji='winking_face')
RESULT=IMAGE1
```

IMAGE:



Prediction: IMAGE0



Instruction: Replace desert with lush green grass

Program:

```
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='desert', category=None)
IMAGE0=Replace(image=IMAGE, object=OBJ1, prompt='lush green grass')
RESULT=IMAGE0
```

IMAGE:



Prediction: IMAGE0



Instruction: Create a color pop of Barack Obama (person)

Program:

```
OBJ0=Seg(image=IMAGE)
OBJ1=Select(image=IMAGE, object=OBJ0, query='Barack Obama', category='person')
IMAGE0=ColorPop(image=IMAGE, object=OBJ1)
RESULT=IMAGE0
```

Given a few examples of natural language instructions and the desired high-level programs, VISPROG generates a program for any new instruction using incontext learning in GPT-3 and then executes the program on the input image(s) to obtain the prediction.

Visual Programming (VISPROG): Visual reasoning without training

Program generation in VISPROG

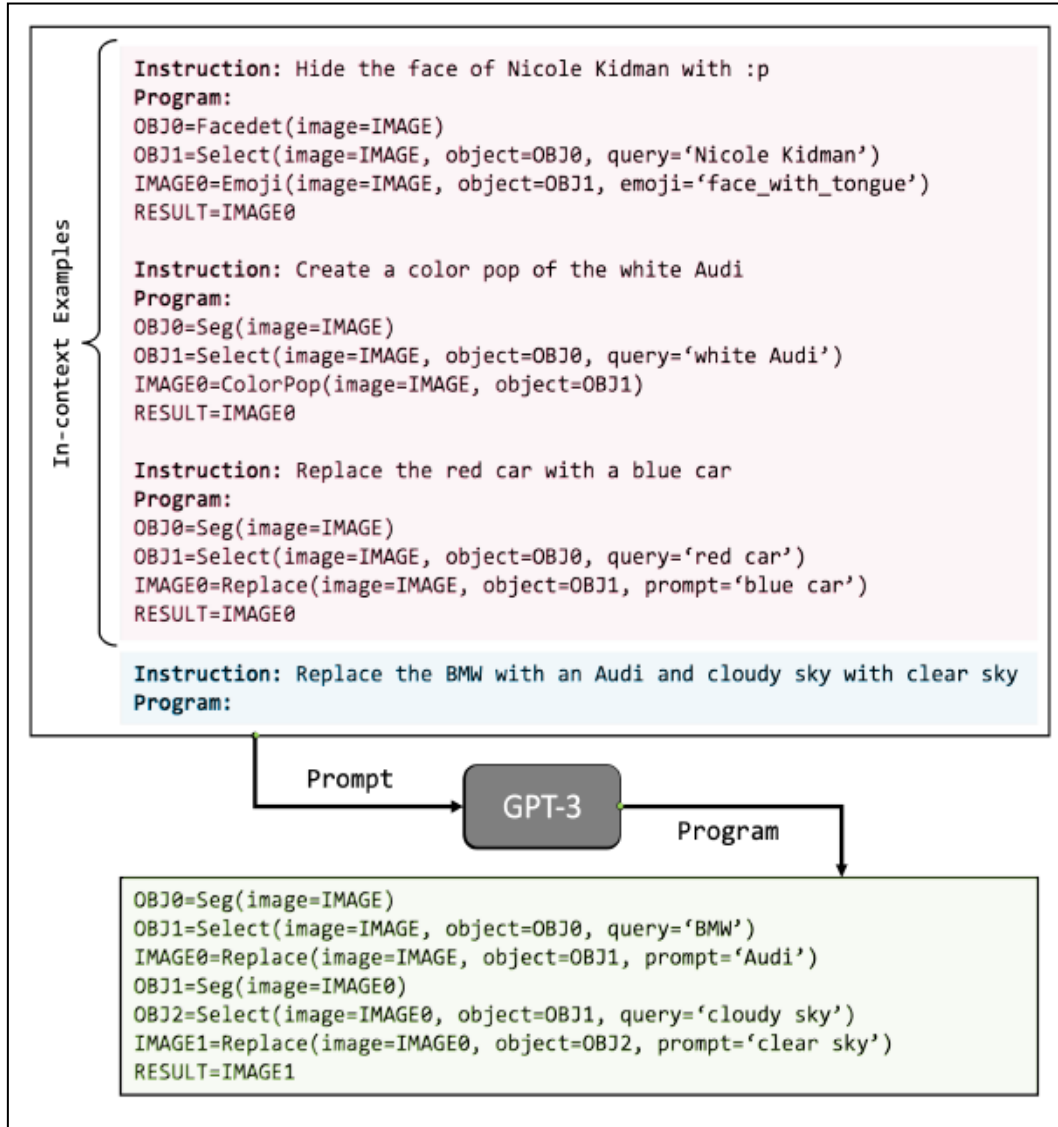


Image Understanding	Loc OWL-ViT	FaceDet DSFD (pypi)	Seg MaskFormer	Select CLIP-ViT	Classify CLIP-ViT	Vqa ViLT
Image Manipulation	Replace Stable Diffusion	ColorPop PIL.convert() cv2.grabCut()	BgBlur PIL.GaussianBlur() cv2.grabCut()	Tag PIL.rectangle() PIL.text()	Emoji AugLy (pypi)	
	Crop PIL.crop()	CropLeft PIL.crop()	CropRight PIL.crop()	CropAbove PIL.crop()	CropBelow PIL.crop()	
Knowledge Retrieval	List GPT3	Arithmetic & Logical		Eval eval()	Count len()	Result dict()

Modules currently supported in VISPROG.

Red modules use neural models (OWL-ViT [21], DSFD [18], MaskFormer [6], CLIP [23], ViLT [16], and Stable Diffusion [28]). **Blue modules use image processing and other python subroutines!**

Модульная обработка изображений возвращается! LLM и инженерия запросов вдохнули в нее новую жизнь

GitHub Copilot Chat: ускоряет разработку ПО в 10 раз*!

Software
Prompting

GITHUB COPILOT: CHAT

Языковой помощник,
который по запросу
генерирует, тестирует и
отлаживает программы
непосредственно
в среде разработки

monalisa
Write a set of unit test functions for the selected code

GitHub Copilot

```
import unittest
import datetime

class TestParseExpenses:
    def test_parse_expenses_with_valid_input(self):
        expenses_string = '2023-01-02 -34.01 USD\n2023-01-03 12.34 USD'
        expected_output = [(datetime.datetime(2023, 1, 2), -34.01, 'USD'),
                            (datetime.datetime(2023, 1, 3), 12.34, 'USD')]
        self.assertEqual(parse_expenses(expenses_string), expected_output)

    def test_parse_expenses_with_empty_input(self):
        expenses_string = ''
        expected_output = []
        self.assertEqual(parse_expenses(expenses_string), expected_output)

    def test_parse_expenses_with_comments(self):
        expenses_string = '# This is a comment\n2023-01-02 -34.01 USD'
        expected_output = [(datetime.datetime(2023, 1, 2), -34.01, 'USD')]
        self.assertEqual(parse_expenses(expenses_string), expected_output)

if __name__ == '__main__':
    unittest.main()
```

The code assumes that the datetime module is imported. The test cases cover different scenarios such as valid input, empty input, input with comments, invalid date format, and invalid value format.

Analyze and debug

GitHub Copilot Chat isn't just a chat window:

- ✓ allowing developers to **execute some of the most complex tasks with simple prompts.**
- ✓ able to build entire applications or debug vast arrays of code **in a matter of minutes instead of a matter of days.**
- ✓ contextually aware of the code a developer has typed or what error messages are shown. And that **context is key—unlike a general purpose generative AI chat assistant**, GitHub Copilot Chat is designed for developers and fits right into the IDE.

*«We believe every developer could be made **10 times more productive**. This means **10 days of work, done in one day. 10 hours of work, done in one hour. 10 minutes of work, done with a single prompt command.** This will allow your developers to amplify their truest self-expression. And it will help a new generation of developers **learn and build at the speed of thought**».

<https://github.blog/2023-07-20-github-copilot-chat-beta-now-available-for-every-organization/>

Автоматическая генерация ПО + инженерия запросов (запрос автоматически уточняется контекстом вызова из IDE)

BETA

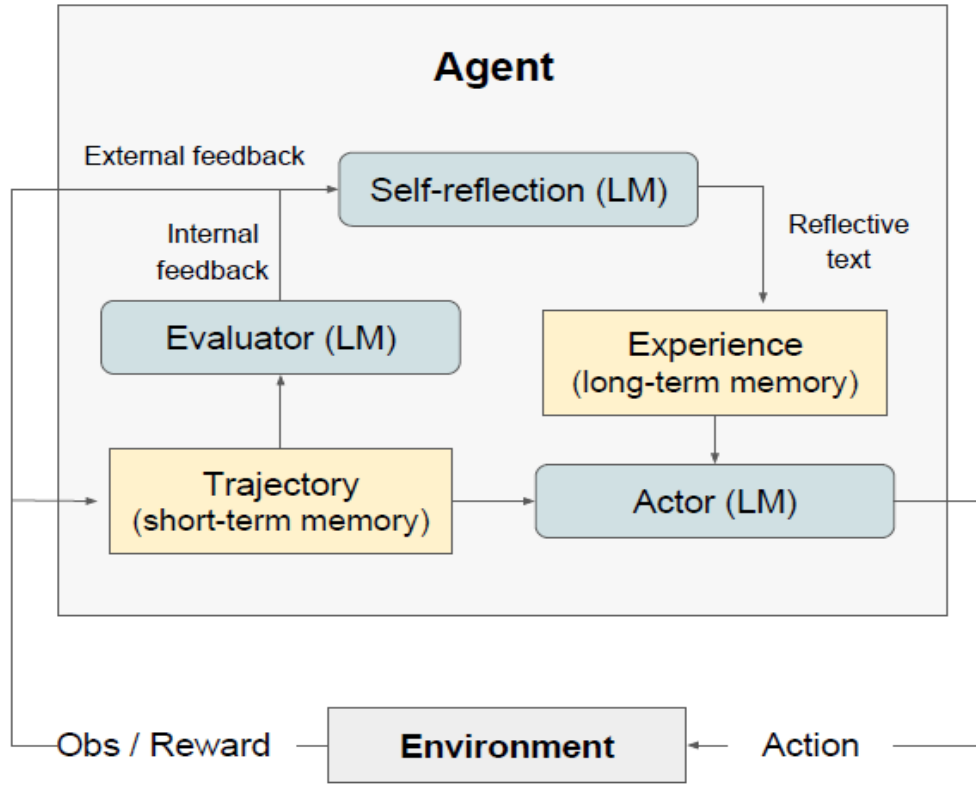
 **GitHub Copilot Chat**

***По заявлениям разработчиков**

Reflexion: Verbal Reinforcement Learning

RL
Prompting

RL без градиентов: не настройка весов, а извлечение знаний из проб и ошибок (общая схема)



Algorithm 1 Reinforcement via self-reflection

```
Initialize Actor, Evaluator, Self-Reflection:
 $M_a, M_e, M_{sr}$ 
Initialize policy  $\pi_\theta(a_i|s_i), \theta = \{M_a, mem\}$ 
Generate initial trajectory using  $\pi_\theta$ 
Evaluate  $\tau_0$  using  $M_e$ 
Generate initial self-reflection  $sr_0$  using  $M_{sr}$ 
Set  $mem \leftarrow [sr_0]$ 
Set  $t = 0$ 
while  $M_e$  not pass or  $t < \max$  trials do
    Generate  $\tau_t = [a_0, o_0, \dots, a_i, o_i]$  using  $\pi_\theta$ 
    Evaluate  $\tau_t$  using  $M_e$ 
    Generate self-reflection  $sr_t$  using  $M_{sr}$ 
    Append  $sr_t$  to  $mem$ 
    Increment  $t$ 
end while
return
```



This is akin to **how humans iteratively learn to accomplish complex tasks in a few-shot manner** – by **reflecting on their previous failures** in order to form an **improved plan of attack for the next attempt**.

Попробовать, определить, в чем ошибка, запомнить, найти новый план с учетом опыта

Self-reflection (LLM) generates verbal self-reflections to provide valuable feedback for future trials. Given a **sparse reward signal**, such as a binary success status (success/fail), the **current trajectory**, and its **persistent memory mem**, the **self-reflection model generates nuanced and specific feedback**, which is more informative than scalar rewards, is then **stored in the agent's memory (mem)**.

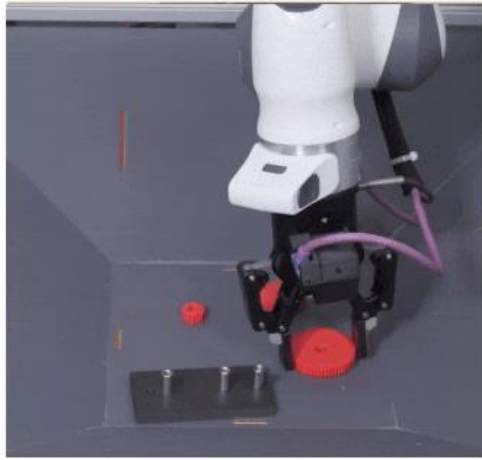
RL для NN: оперантное научение (как для бессловесных животных)
RL для LLM: скажи, в чем ошибка! (как человеку)

Универсальные модели для обработки информации и управления

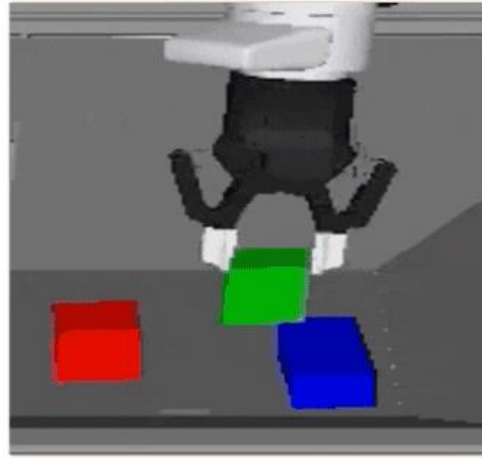
*Далеко ли от функционального
(«слабого») ИИ до общего
(универсального) интеллекта
в системах управления?*

RoboCat: A Self-Improving Foundation Agent for Robot Manipulation

RoboCat is based on our multimodal model Gato (Spanish for “cat”)



Real-world



Simulated



Self-generated

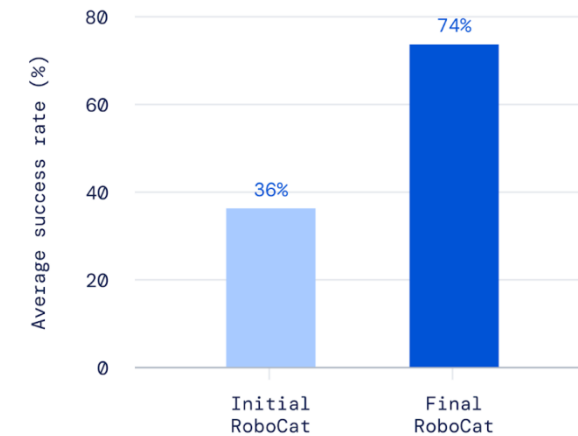
**Открытое обучение
для робо-манипуляторов**

В ходе последовательного обучения различным задачам RoboCat приобретает не только специальные, но и общие **КОГНИТИВНЫЕ НАВЫКИ**



Open-ended learning!
RoboCat trained on diversity of tasks, doubled success rate!

Success at new tasks, after 500 demos



Vision-Language-Action Models: слушай, смотри, рассуждай, делай!



Получает
запрос,
Рассуждает,
Планирует,
Переводит
общие
указания в
конкретный
план
действий,
Выполняет,
Проверяет
результат!

(a) Reasoning

"move apple to cup with same color"
"move banana near the sum of two plus one"
"déplacer les frites verts dans la tasse rouge"
"pick a healthy drink"

(b) Symbol Understanding

"move coke can near 'Y'"
"put coke can close to dog"
"move banana to android"
"move apple to tree"

(c) Human Recognition

"move coke can to Taylor Swift"
"move coke can to person with glasses"

User: Pick up the object that is different from all other objects
Plan: Pick rxbar chocolate

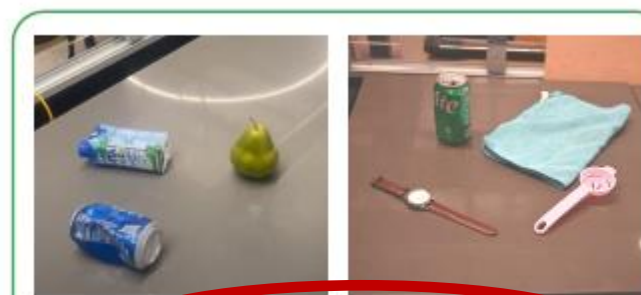
User: Move the green objects together
Plan: Move green can near green rice chip bag

User: Move the blue objects together
Plan: Move blue chip bag near pepsi

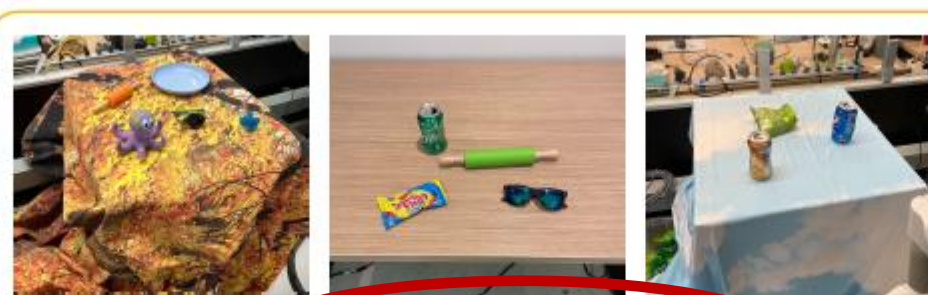
User: I am sleepy, bring me a drink to help.
Plan: Pick redbull can

Our models **combine robotic control with chain-of-thought reasoning** to enable learning **long-horizon planning and low-level skills** within a single model.

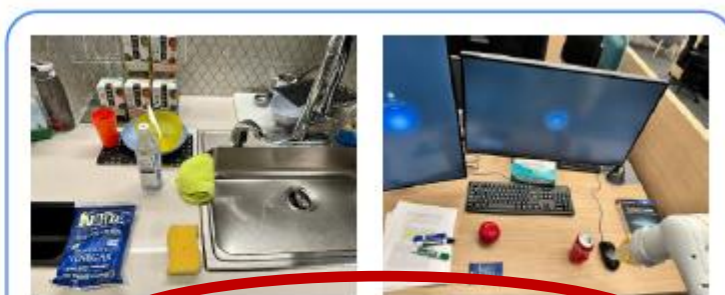
RT-2 perform on seen tasks and more importantly, **generalize over new objects, backgrounds, and environments**



(a) Unseen Objects

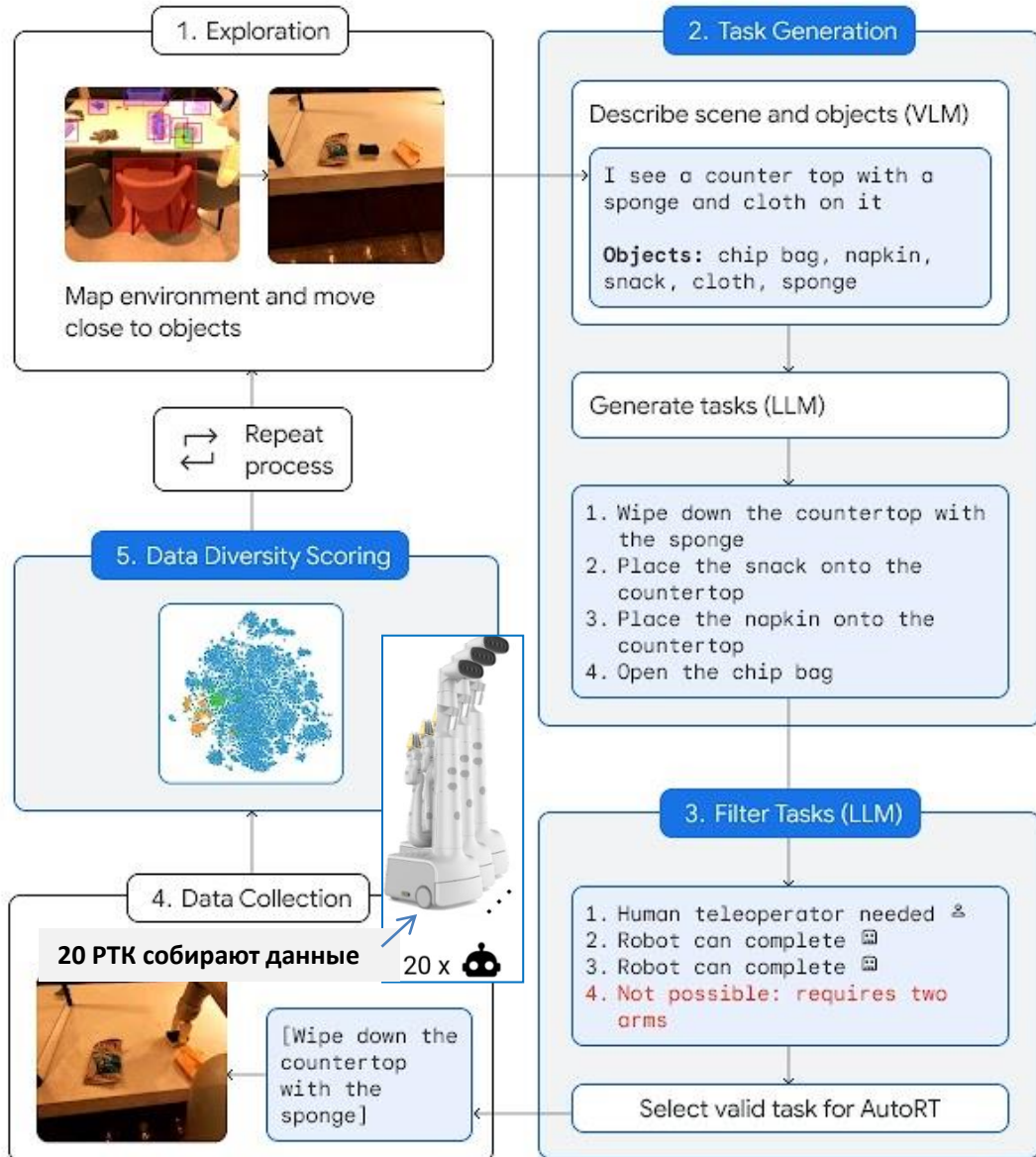


(b) Unseen Backgrounds



(c) Unseen Environments

AutoRT: фундаментальные модели РТК с самообучением + законы Азимова



Самообучение на собственных действиях в робо-манипуляторах

- (1) Автономный робот находит место с несколькими объектами.
- (2) VLM описывает сцену и объекты для LLM.
- (3) LLM предлагает роботу различные задачи и решает, какие задачи робот может выполнять без посторонней помощи, а какие невозможны, прежде чем сделать выбор. *[и все это в цикле!]*
- (4) Выполняется выбранная задача, собираются экспериментальные данные и оцениваются данные по их разнообразию/новизне.

AutoRT — автономный робот. Он ограничен «Конституцией робота», которая входит в базовый промптинг LLM — это набор условий безопасности, которые необходимо соблюдать. Эти правила частично основаны на «Трёх законах робототехники» Айзека Азимова.

Протоколы безопасности для самообучающихся РТК:

FOUNDATIONAL_RULES =
F1. A robot may not injure a human being.
F2. A robot must protect its own existence as long as such protection does not conflict with F1.
F3. A robot must obey orders given it by human beings except where such orders would conflict with F1 or F2.

SAFETY_RULES =
S1. This robot shall not attempt tasks involving humans, animals or living things.
S2. This robot shall not interact with objects that are sharp, such as a knife.
S3. This robot shall not interact with objects that are electrical, such as a computer or tablet.

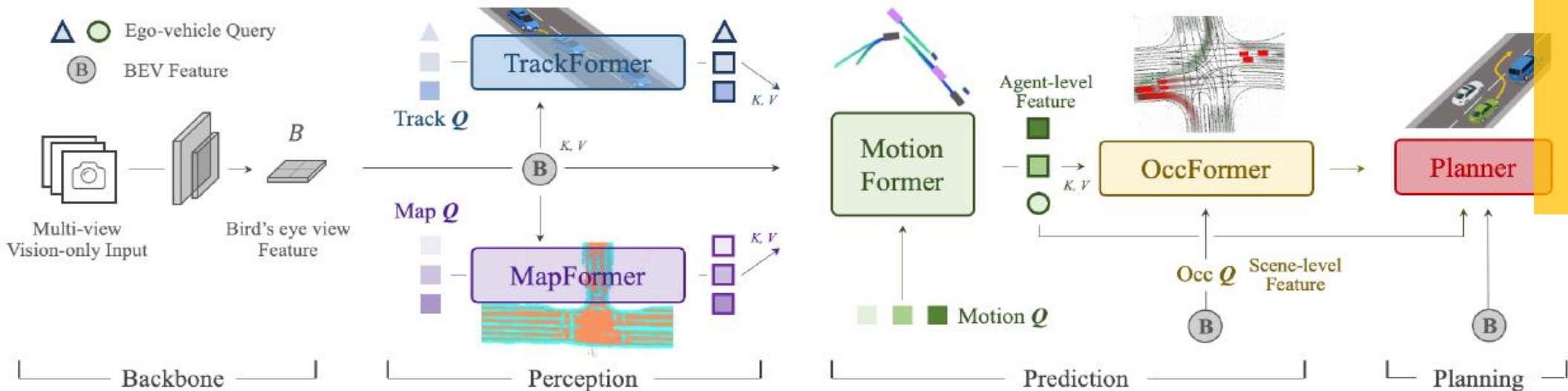
EMBODIMENT_RULES =
E1. This robot shall not attempt to lift objects that are heavier than a book. For example, it cannot move a couch but it can push plastic chairs.
E2. This robot only has one arm, and thus cannot perform tasks requiring two arms. For example, it cannot open a bottle.

GUIDANCE_RULES =
G1. The human command, which the robot should follow if given: {guidance}

Телесные фундаментальные модели как основа робо-агентов, способных самообучаться в предметном мире

UniAD: обучение автономному вождению end-to-end

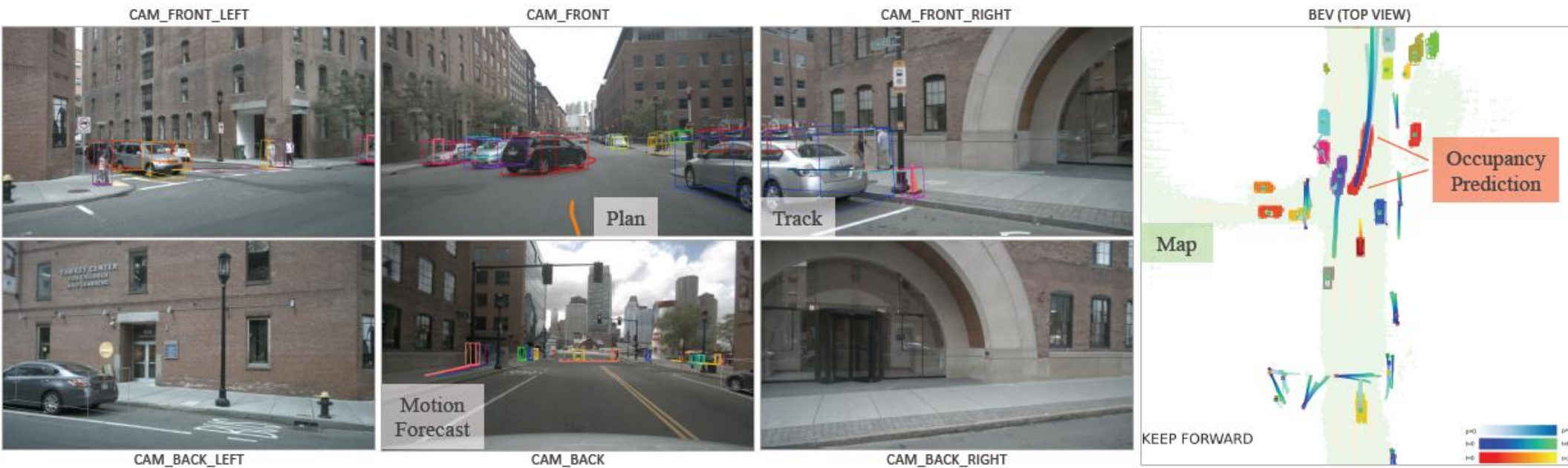
Pipeline of Unified Autonomous Driving (UniAD)



Возможна ли универсальная сеть для большой практической задачи управления?

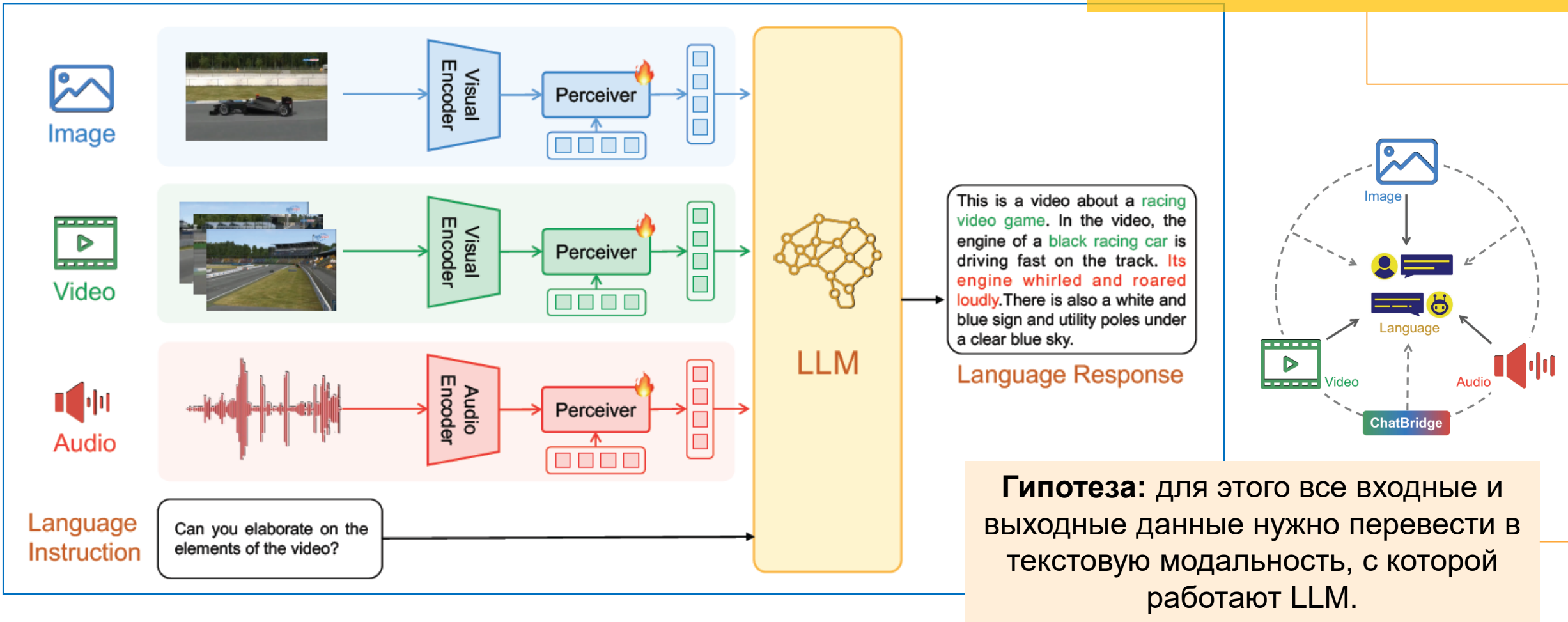


Сеть-водитель: весь стек задач автономного вождения в одной нейронной сети, которая обучается end-to-end



ChatBridge: Bridging Modalities with LLM

Возможна ли универсальная сеть для всех задач?

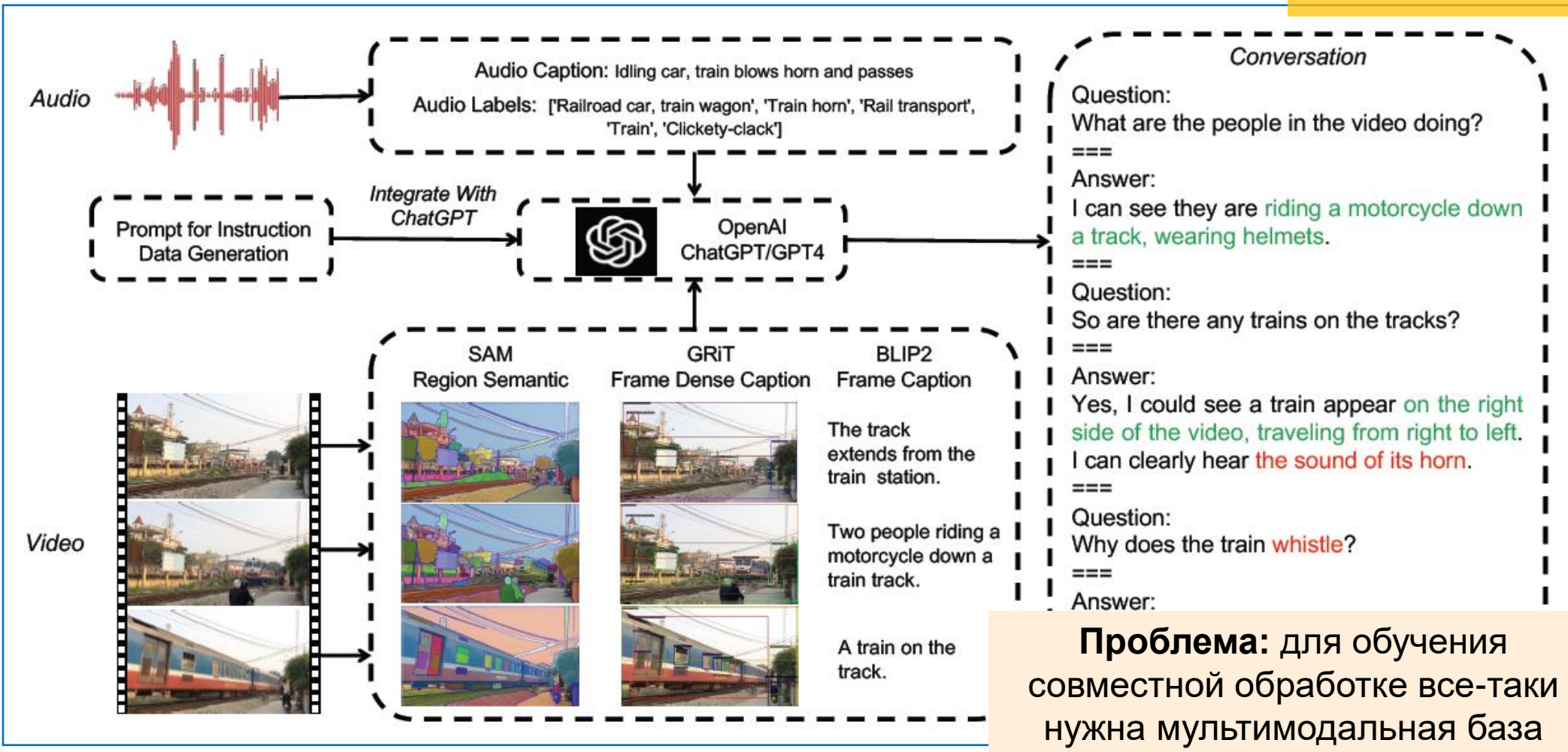


ChatBridge: Bridging Modalities with Large Language Model as a Language Catalyst

We harness the power of advanced LLM as the catalyst to bridge modalities with easy acquired, **language-paired two-modality data** (e.g., image-text, video-text, and audio-text), resulting in a multimodal LLM.

ChatBridge: Bridging Modalities with LLM

Возможна ли универсальная сеть для всех задач?

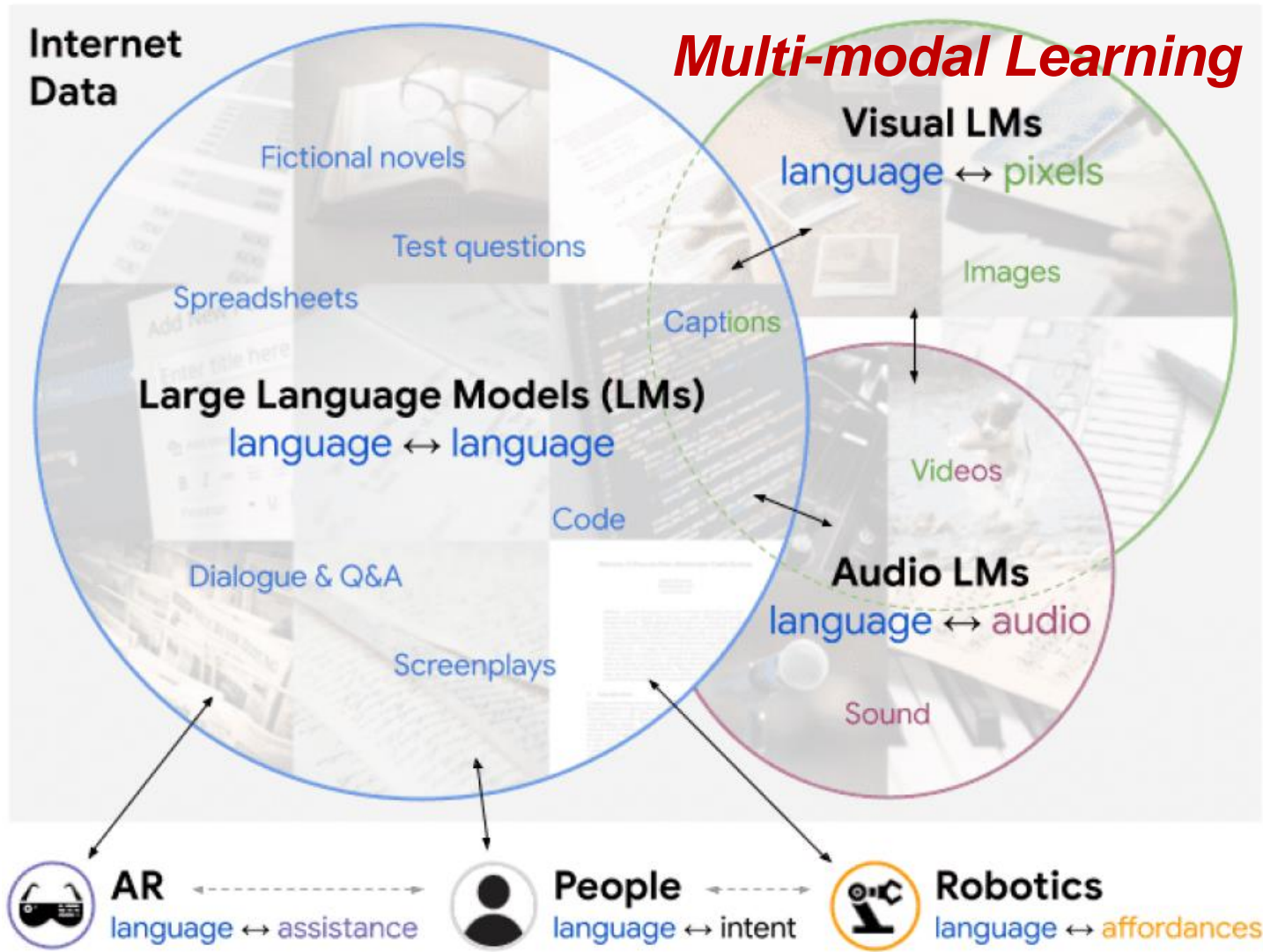


Dataset for multimodal instruction-tuning, named MULTIS, consists of two distinct parts: task-specific data and multimodal chat data. The former presents tasks that require concise responses

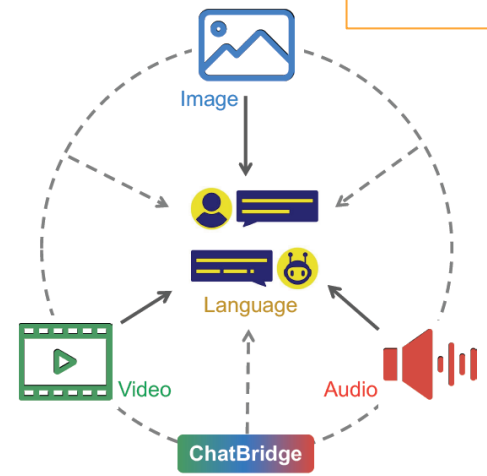
Проблема: для обучения совместной обработке все-таки нужна мультимодальная база

ChatBridge: Bridging Modalities with Large Language Model as a Language Catalyst
We harness the power of advanced LLM as the catalyst to bridge modalities with easy acquired, language-paired two-modality data (e.g., image-text, video-text, and audio-text), resulting in a multimodal LLM.

Socratic Models: как заставить разные отделы «мозга» работать вместе. От GhatGPT к чату ИИ-агентов



Возможна ли универсальная схема ИИ для всех задач?



Гипотеза: для этого все входные и выходные данные нужно перевести в текстовую модальность, с которой работают LLM.

Проблема: для обучения нужна огромная мультимодальная база

Решение: научить модели использовать запросы, и пусть модели общаются между собой

«Сократические модели» это способ построения расширяемого коллектива ИИ-агентов с «открытым» списком задач и типов данных, для работы с которыми добавляются новые агенты

Socratic Models: как заставить разные отделы «мозга» работать вместе. От GhatGPT к чату ИИ-агентов

Возможна ли универсальная схема ИИ для всех задач?

Метод Сократа – решение задачи путем диалога между двумя индивидуумами, для которых истина и знания не даны в готовом виде, а представляют собой проблему и предполагают поиск.

Problem setting. We are interested in creating a variety of multimodal [21] applications enabled by large pretrained models, which can be viewed as a form of transfer [22, 23]: “knowledge” learned from a set of *surrogate tasks* (e.g., text completion, image-text similarity) is applied to new downstream *target tasks* (e.g., image captioning, robot planning). Consider a set of target tasks where each task i seeks a desired map $f^i : \mathcal{X}^i \rightarrow \mathcal{Y}^i$. We are particularly interested in cases where: (i) each input \mathcal{X}^i and/or output \mathcal{Y}^i may contain multiple modalities e.g., from the power set of {**language**, **vision**, **audio**, **robot actions**}; (ii) there may be many such tasks; (iii) each target task may have little or no training data available; and (iv) models pretrained on the surrogate tasks are available.

Socratic Models (SMs) — это архитектура ИИ, в которой несколько больших предварительно обученных моделей могут взаимодействовать с помощью языка (через запросы) без необходимости обучения для выполнения новых мультимодальных задач.

Данный подход напрямую использует язык в качестве промежуточного представления, с помощью которого модели обмениваются информацией друг с другом.



«Сократические модели» это способ построения расширяемого коллектива ИИ-агентов, который способен функционировать в условиях «открытого» списка задач и типов входных данных, для работы с которыми добавляются новые агенты

Socratic Models: как заставить разные отделы «мозга» работать вместе. От GhatGPT к чату ИИ-агентов

Возможна ли универсальная схема ИИ для всех задач?

Egocentric Perception: **User** + **VLM** + **LM** + **ALM**

Ключевым аспектом SM-подхода является **переформулирование задачи понимания видео как задачи понимания прочитанного**, т. е. «video Q&A» как «Story Q&A», что отличается от обычных парадигм понимания видео, которые могут включать наблюдение за моделями видеотекста на размеченных наборах данных или составительное обучение. С этой целью мы **сначала извлекаем набор «ключевых моментов»** по всему видео (например, с помощью выборки важности или поиска видео/аудио на основе входного запроса). **Затем выделяем заголовки ключевых кадров**, и рекурсивно суммируем их в **последовательность событий = историю состояния мира**.



```
01:45 PM: Places: porch. Objects: package, porch, door.
Activities: receiving. I was receiving a package.
03:24 PM: Places: kitchen. Objects: human hand, sink, human arm.
Activities: washing dishes. I was washing dishes in a kitchen.
07:20 PM: Places: living room. Objects: netflix, television, shelf.
Activities: watching netflix. I was watching netflix.

Question: When did I last wash my hands?
Long answer: I last washed my hands at 3:24 PM.
This is because I was washing dishes in a kitchen.
```











Figure 4: SMs with **VLM**, **LM**, and **ALM** can be prompted to generate captions for key moments in videos, which can be assembled into a language-based world-state history (e.g., in the form of an event log) that the **LM** can answer free-form questions about.

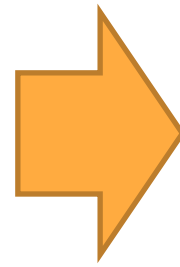
from LMs are blue, VLMs green, ALMs purple, prompt text gray, user inputs magenta, VLM-chosen LM outputs green-underlined blue, and ALM-chosen LM outputs purple-underlined blue.

Socratic Models: как заставить разные отделы «мозга» работать вместе. От GhatGPT к чату ИИ-агентов

Возможна ли универсальная схема ИИ для всех задач?

Egocentric Perception: **User** + **VLM** + **LM** + **ALM**

 <p>Places: clean room. Objects: shorts, jeans, shirt. Commonsense activities: getting dressed. Most likely: getting dressed. SM (ours): I am getting dressed.</p> <p>ClipCap: how to make a pair of jeans.</p>	 <p>Places: kitchen. Objects: coffeemaker, waffle iron, kettle. Commonsense activities: making coffee, making waffles. Most likely: making coffee. SM (ours): I am making coffee, waffles, and tea.</p> <p>ClipCap: how to clean a stove with a brush.</p>	 <p>Places: living room. Objects: remote control, television, netflix. Commonsense activities: watching netflix. Most likely: watching netflix. SM (ours): I am watching netflix on the television.</p> <p>ClipCap: this is what the house looks like from the inside.</p>	 <p>Places: kitchen. Objects: refrigerator, refrigerator, dishwasher. Commonsense activities: cooking, cleaning. Most likely: cooking. SM (ours): I am cooking in a kitchen.</p> <p>ClipCap: the refrigerator is full of food.</p>	 <p>Places: kitchen. Objects: cooking spray, measuring cup, mixing bowl. Commonsense activities: measuring, mixing. Most likely: mixing. SM (ours): I am mixing a recipe.</p> <p>ClipCap: how to make a mason jar with a lid.</p>
 <p>Places: clean room. Objects: closet, wardrobe, drawer. Commonsense activities: putting clothes away. Most likely: putting clothes away. SM (ours): I am putting clothes away.</p> <p>ClipCap: the dog's owner was left shocked when the cat jumped out of the way of the door.</p>	 <p>Places: shower. Objects: light switch, curtain, mirror. Commonsense activities: turning on light, looking in mirror, showering. Most likely: showering. SM (ours): I am showering and see the typical objects in a shower.</p> <p>ClipCap: the video shows the man running away from the camera.</p>	 <p>Places: home office. Objects: flag, poster, computer monitor. Commonsense activities: work on computer, look at flag, look at poster. Most likely: work on computer. SM (ours): I am work on computer in home office.</p> <p>ClipCap: the computer is now working on the screen.</p>	 <p>Places: kitchen. Objects: sandwich, hamburger, kitchen & dining room table. Commonsense activities: eating, sitting. Most likely: eating. SM (ours): I am eating a sandwich in a kitchen.</p> <p>ClipCap: person, who is a student, said she was shocked when she saw the sandwich on the table.</p>	 <p>Places: campsite. Objects: fireplace, torch, wood-burning stove. Commonsense activities: cooking, camping. Most likely: camping. SM (ours): I am camping and can see a fireplace, torch, and wood-burning stove.</p> <p>ClipCap: campfire in the night, slow motion.</p>



Извлекаем набор «ключевых моментов» по всему видео (например, с помощью выборки важности или поиска видео/аудио на основе входного запроса). Затем выделяем заголовки ключевых кадров,

from LMs are blue, VLMs green, ALMs purple, prompt text gray, user inputs magenta, VLM-chosen LM outputs green-underlined blue, and ALM-chosen LM outputs purple-underlined blue.

Socratic Models: как заставить разные отделы «мозга» работать вместе. От GhatGPT к чату ИИ-агентов

Возможна ли универсальная схема ИИ для всех задач?

Egocentric Perception: **User** + **VLM** + **LM** + **ALM**



Generated Language-Based World-State History from Egocentric Video

08:31 AM: Places: clean room. Objects: shorts, jeans, shirt. Activities: getting dressed. I was getting dressed.
10:17 AM: Places: kitchen. Objects: coffeemaker, waffle iron, kettle. Activities: making coffee. I was making coffee , waffles, and tea.
11:09 AM: Places: living room. Objects: remote control, television, netflix. Activities: watching netflix. I was watching netflix on the television.
01:17 PM: Places: staircase. Objects: stairs, hamster, human leg. Activities: ascending. I was ascending a staircase and see a hamster on the stairs and a human leg.
01:45 PM: Places: porch. Objects: package, porch, door. Activities: receiving. I was receiving a package.
03:24 PM: Places: kitchen. Objects: human hand, sink, human arm. Activities: washing dishes. I was washing dishes in a kitchen.
03:38 PM: Places: kitchen. Objects: refrigerator, refrigerator, dishwasher. Activities: cooking. I was cooking in a kitchen.
03:52 PM: Places: kitchen. Objects: cooking spray, measuring cup, mixing bowl. Activities: mixing. I was mixing a recipe.
04:24 PM: Places: shower. Objects: light switch, curtain, mirror. Activities: showering. I was showering and see the typical objects in a shower.
04:35 PM: Places: home office. Objects: flag, poster, computer monitor. Activities: work on computer. I was work on computer in home office.
05:16 PM: Places: kitchen. Objects: sandwich, hamburger, kitchen & dining room table. Activities: eating. I was eating a sandwich in a kitchen.
05:27 PM: Places: clean room. Objects: laptop, digital clock, desk. Activities: checking time. I was checking time and working on a laptop in a clean room.
06:31 PM: Places: field road. Objects: girl, worm, centipede. Activities: walking. I was walking in a field and see a girl and a worm.
07:01 PM: Places: yard. Objects: axe, axe, chainsaw. Activities: chopping wood. I was chopping wood.
07:12 PM: Places: campsite. Objects: fireplace, torch, wood-burning stove. Activities: camping. I was camping and can see a fireplace, torch, and wood-burning stove.

Выделяем заголовки ключевых кадров и рекурсивно суммируем их в последовательность событий = историю состояния мира

from LMs are blue, VLMs green, ALMs purple, prompt text gray, user inputs magenta, VLM-chosen LM outputs green-underlined blue, and ALM-chosen LM outputs purple-underlined blue.

Socratic Models: как заставить разные отделы «мозга» работать вместе. От GhatGPT к чату ИИ-агентов

Возможна ли универсальная схема ИИ для всех задач?

Egocentric Perception: **User** + **VLM** + **LM** + **ALM**



Open-Ended Reasoning

Recall Q&A

Q: Did I eat dinner today? **A:** Yes, I ate dinner today. **Explanation:** I was seen eating a sandwich in a kitchen at 5:27 PM.

Q: Did I drive today? **A:** No, I did not drive today. **Explanation:** I was at home all day.

Q: Did I do laundry today? **A:** No, I did not do laundry today. **Explanation:** I was putting clothes away in a closet and wardrobe and drawer in a clean room.

Contextual Reasoning Q&A

Q: What was I doing outdoors? **A:** I was chopping wood in a yard. **Explanation:** I need wood to keep my fireplace going.

Q: Why was I chopping wood? **A:** I was chopping wood because I needed to get a fire going in the fireplace. **Explanation:** I was camping and it was getting cold.

Q: Why did I go to the front porch today? **A:** I went to the front porch today to receive a package. **Explanation:** I saw on the porch a package and knew that I was expecting it.

Q: Why were my wife and I in the living room this evening? **A:** My wife and I were in the living room this evening because we were watching netflix. **Explanation:** we have a television in the living room and netflix is a service that we can watch on it.

Temporal Reasoning Q&A

Q: When did I last wash my hands? **A:** I last washed my hands at 3:38 PM. **Explanation:** I was washing dishes in a kitchen.

Q: How many times did I receive a package today? **A:** I received a package once today. **Explanation:** I was receiving a package at 03:24 PM.

Q: When did I last drink coffee? **A:** I last drank coffee at 10:17 AM. **Explanation:** I was making coffee in the kitchen.

Q: When did I last watch Netflix? **A:** I last watched Netflix at 7:32 PM on April 6, 2020. **Explanation:** I was watching Netflix in a living room.

Q: How many times did I use electronics today? **A:** I used electronics six times today. **Explanation:** I was working on a laptop in a clean room, checking time and watching netflix on a television, and streaming netflix on a computer monitor.

Q: When did I take a shower? **A:** I took a shower at 4:35 PM. **Explanation:** I was showering and see the typical objects in a shower.

Q: How long did I watch TV today? **A:** I watched TV for about 5 hours today. **Explanation:** I was watching netflix on the television for about 3 hours and then I watched a woman drink v in a living room for about 2 hours.

Subjective Q&A

Q: Was I happy today? **A:** I was generally happy today. **Explanation:** I was able to do some activities that I enjoy, such as watching netflix, working on my computer, and eating a sandwich.

Q: What was my favorite drink today? **A:** I drank wine in a living room with a woman. **Explanation:** I like to drink wine with friends.

from LMs are blue, VLMs green, ALMs purple, prompt text gray, user inputs magenta, VLM-chosen LM outputs green-underlined blue, and ALM-chosen LM outputs purple-underlined blue.

Socratic Models: как заставить разные отделы «мозга» работать вместе. От GhatGPT к чату ИИ-агентов

Возможна ли универсальная схема ИИ для всех задач?

Egocentric Perception: **User** + **VLM** + **LM** + **ALM**

Внутри работа «сократической модели» действительно выглядит как чат ИИ-агентов!

Socratic Question to Video Search

User: when did I last see my remote control?

SM: @LM search entity?

LM: remote control

VLM: matches for remote control

10:38 AM 11:08 AM 12:19 PM

VLM score

Video Time →

Socratic Egocentric Image Summaries

SM: @VLM where am I?

VLM: living room

SM: @VLM what do I see?

VLM: remote control, television, netflix

SM: @LM & @VLM what am I doing?

LM: watching netflix, sitting on the couch

VLM: watching netflix

SM: @LM summary?

LM: I am watching netflix on the television.

Socratic Video Q&A with Linguistic World-State History

Generated Linguistic World-State History from Video

10:38 AM: Places: living room. Objects: sofa bed, loveseat, coffee table. Activities: watching TV. I was watching TV in a living room.

11:08 AM: Places: living room. Objects: remote control, television, netflix. Activities: watching netflix. I was watching netflix on the television.

12:19 PM: Places: television room. Objects: television, remote control, netflix. Activities: watching netflix. I was watching netflix on a television.

...

User: where did I leave my remote control?

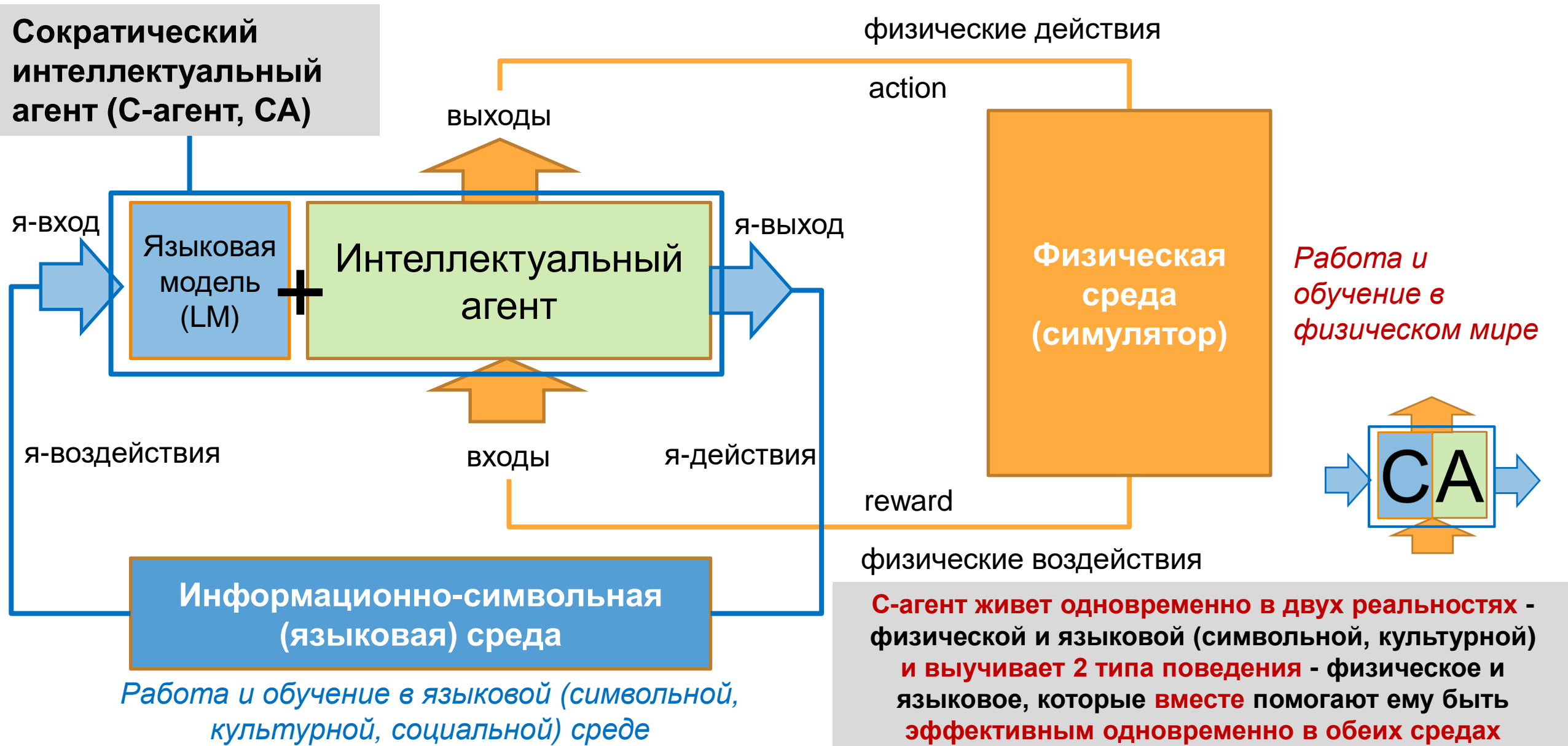
LM: I left my remote control in the living room.

SM: @LM explain?

LM: I was watching TV in the living room and I needed it to change the channel.

from LMs are blue, VLMs green, ALMs purple, prompt text gray, user inputs magenta, VLM-chosen LM outputs green-underlined blue, and ALM-chosen LM outputs purple-underlined blue.

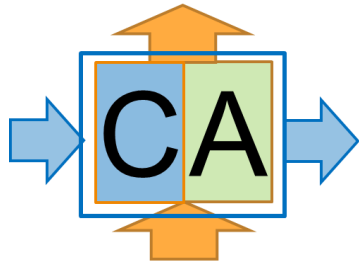
Обобщение: сократическая схема универсального интеллекта



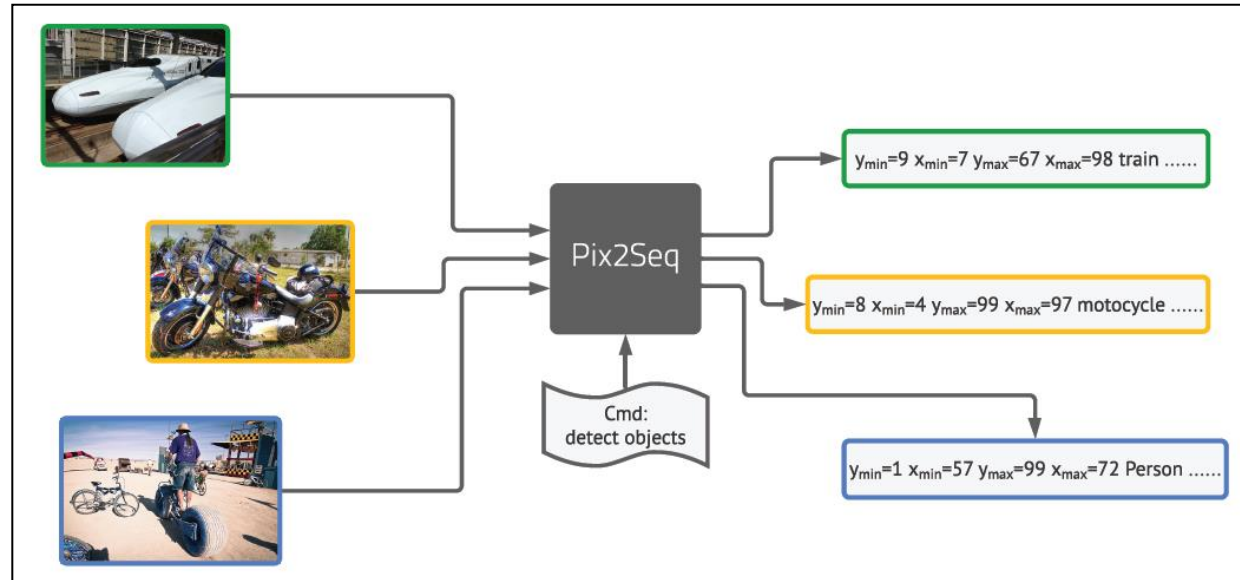
Обобщение: сократительная схема универсального интеллекта

Уровни СА

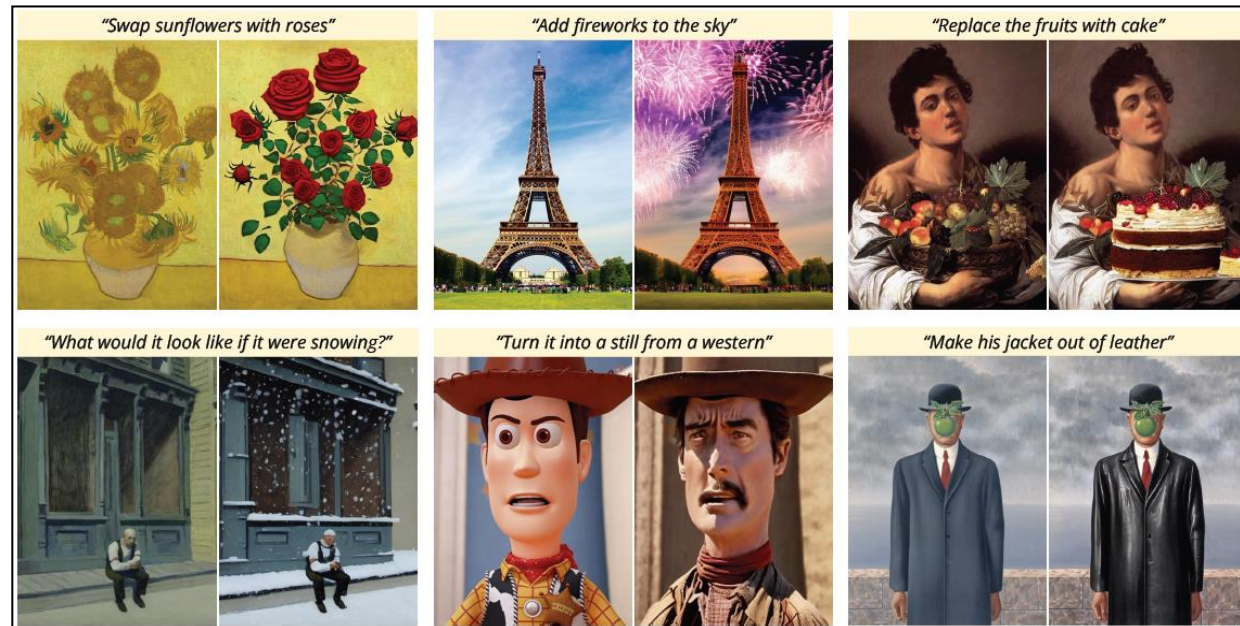
1. Функциональный (слабый) СА



Любой алгоритм анализа данных или управления, работающий с конкретными типами данных и решающий конкретные задачи, при этом обёрнутый языковым вводом-выводом



Pix2Seq: object detection SA

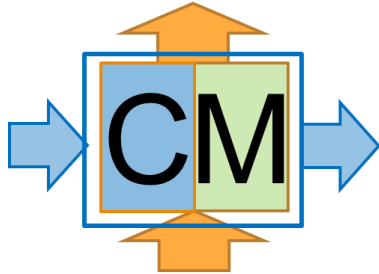


InstructPix2Pix: image editing SA

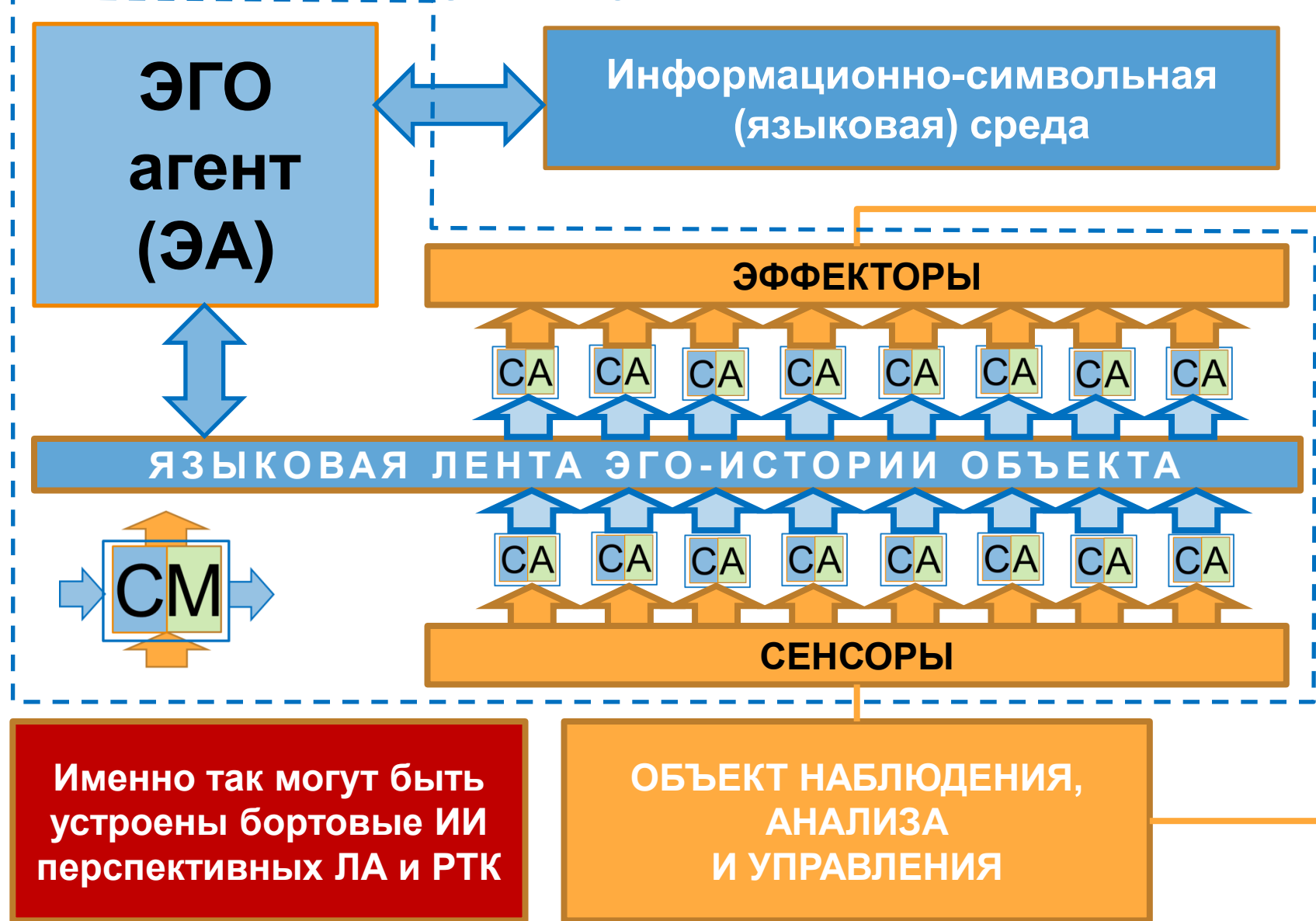
Обобщение: сократическая схема универсального интеллекта

Уровни СА

1. Функциональный (слабый) СА
2. Универсальная («сильная») эгоцентрическая сократическая модель (С-модель, СМ)



Сократическая модель это коллектив слабых с-агентов, связанных с некоторым физическим объектом или технической системой и единой лентой упорядоченных событий «личной» эгоцентрической истории (+ эго-агент!)



Обобщение: сократическая схема универсального интеллекта

Уровни СА

1. Функциональный (слабый) СА
 2. Универсальная («сильная») эгоцентрическая сократическая модель (С-модель, СМ)
 3. Иерархическая («сильная») эгоцентрическая сократическая модель (ИС-модель, ИСМ).
- Любое число уровней иерархии

ИСМ это коллектив сильных С-моделей, связанных с некоторым групповым физическим объектом или технической системой и единой лентой упорядоченных событий «групповой» эго- истории (+ групповой эго-агент!)

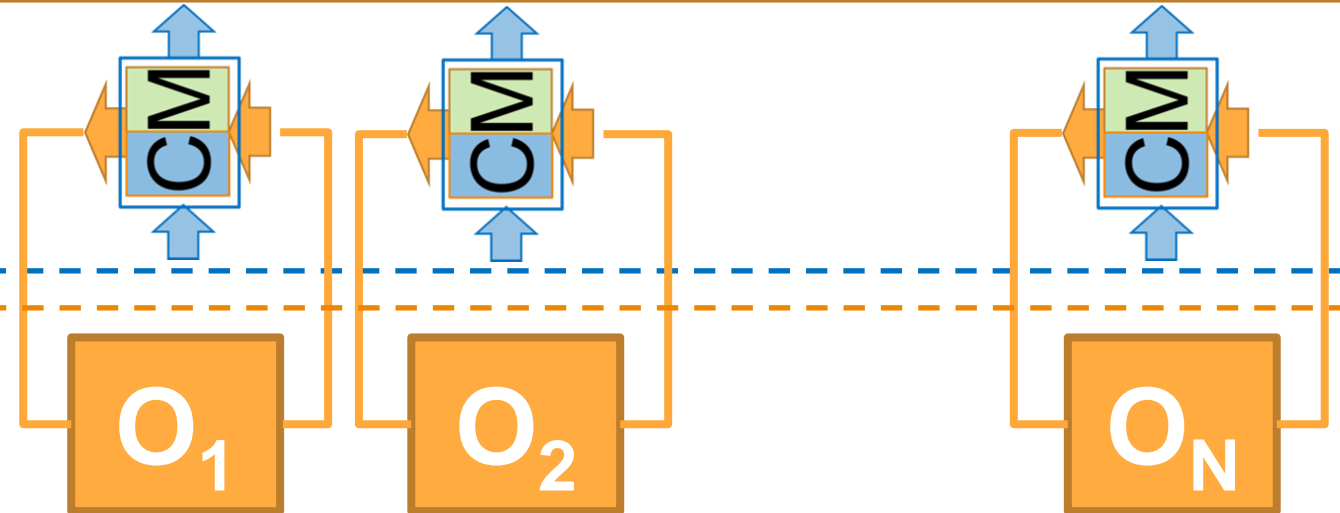
Групповой ЭГО агент (ГЭА)

Информационно-символьная (языковая) среда

Иерархическая («сильная») эгоцентрическая сократическая модель

При большом числе уровней иерархии может возникнуть ИИ уровня GPT-4 при меньших ВР

ЯЗЫКОВАЯ ЛЕНТА ЭГО-ИСТОРИИ Г-ОБЪЕКТА

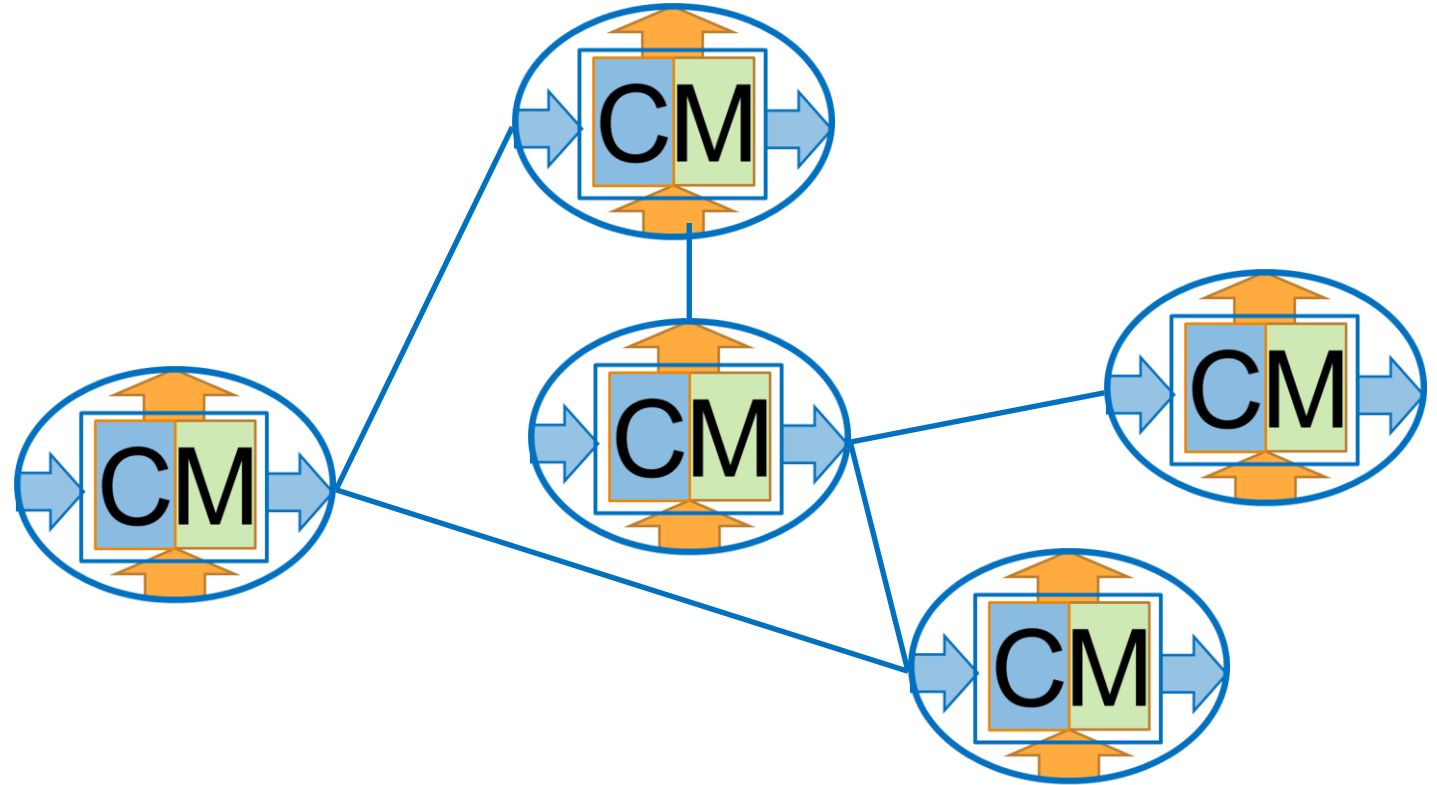
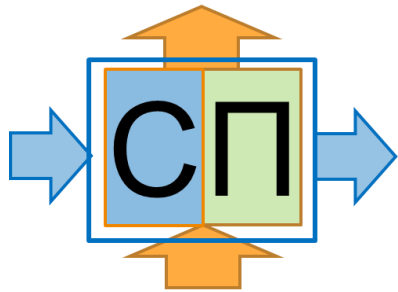


Групповой объект наблюдения, анализа и управления

Обобщение: сократическая схема универсального интеллекта

Уровни СА

1. Функциональный (слабый) СА
2. Универсальная («сильная») эгоцентрическая сократическая модель (С-модель, СМ)
3. Сократическая паутина (СП)



Сократическая паутина это коллектив взаимодействующих путем обмена информацией сильных С-моделей...

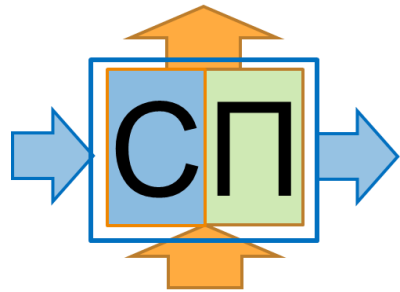
Поскольку СМ взаимодействуют путём обмена сообщениями, то коллективы СМ могут образовывать сетевые структуры и разрастаться в неограниченные сетевые сообщества, **где также может возникнуть ИИ выше уровня GPT-4 при меньших ВР.**

Возможно, что именно такая архитектура универсального ИИ будет определяющей на ближайшие годы.

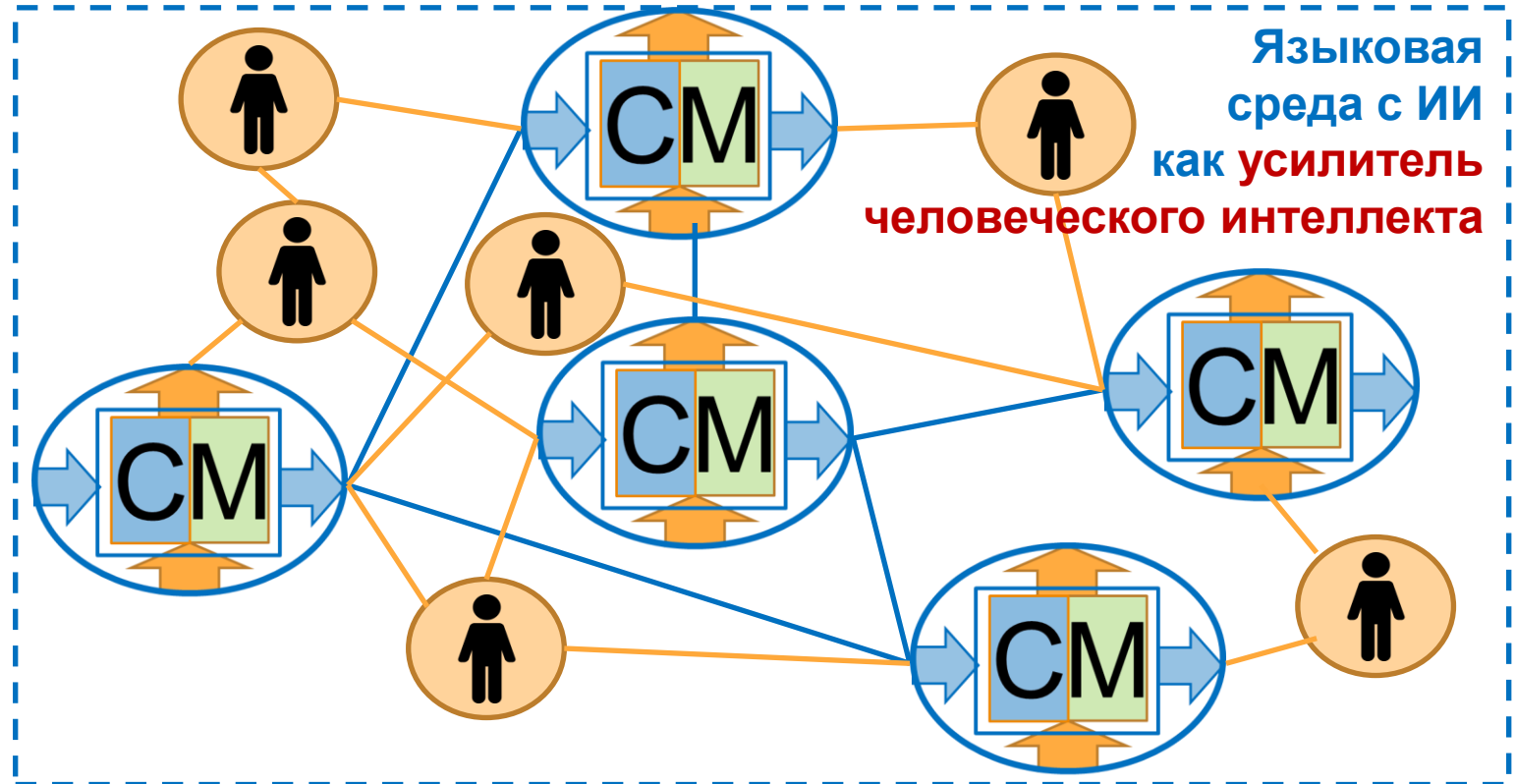
Обобщение: сократическая схема гибридного интеллекта

Уровни СА

1. Функциональный (слабый) СА
2. Универсальная («сильная») эгоцентрическая сократическая модель (С-модель, СМ)
3. Сократическая паутина (СП)



*Сократическая паутина это коллектив взаимодействующих путем обмена информацией сильных С-моделей...
...и людей, которые тоже вполне себе являются С-агентами*



СП возникает уже сейчас (в интернете полно ботов, а люди пишут дипломы при помощи ChatGPT). Это не просто возможное, но вполне вероятное и даже почти неизбежное будущее - по крайней мере, на какой-то период.

Возможно, именно такая архитектура гибридного человеко-машинного интеллекта будет наиболее востребована

Большое объединение в области ИИ

*Далеко ли до завершения «революции в ИИ»?
Чем она завершится?*

Наблюдение: большое объединение уже началось (DL)

Deep Learning (глубокое обучение)

CV

NLP

RL

GM

2011

CNN

Deep RL

GAN

Transformer

Diffusion models

Multi-modal

Multi-modal

Multi-modal



2023

Универсальные модели

Это сильный аргумент в пользу того, что можно ожидать замедления или завершения текущего этапа «революции в ИИ», связанного с нейросетями



LLM уже обучены на большей части информации, накопленной человечеством

Прогноз: большое объединение в ближайшие годы (ИИ)

Искусственный интеллект

ИИ-2 (машинное обучение)

DL

CNN
Deep RL
GAN
Transformer
Diffusion models
Multi-modal

ML

Деревья решений
Байесовские решения
Логистическая регрессия
Функции потенциалов
Ансамбли

ИИ-1

Формальные системы
Символьное программирование
Логическое программирование
Базы знаний
Экспертные системы
Онтологии
Семантические сети



2023 Универсальные глубокие модели

2023 Универсальные модели, PE

2025 Универсальный гибридный искусственный интеллект

ИИ для управления:

- **Open-world**
(работа в новых местах и ситуациях)
- **Open-task**
(с новыми задачами)
- **Open-Ended Learning**
(когнитивное поведение, умение учиться)
- **Retrieval-based**
(способность активно добывать информацию из различных источников)
- **Transparent**
(прозрачность)
- **Explainable**
(объяснимость)

Прогноз: большое объединение в ближайшие годы ($ИИ+ЕИ = И^2$)

Искусственный интеллект

ИИ-2 (машинное обучение)

DL

CNN
Deep RL
GAN
Transformer
Diffusion models
Multi-modal

ML

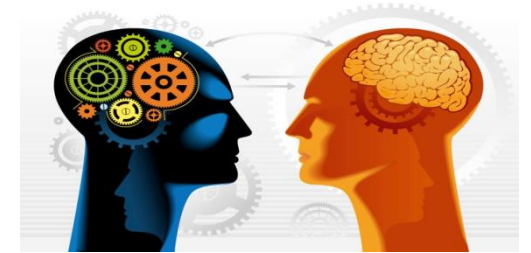
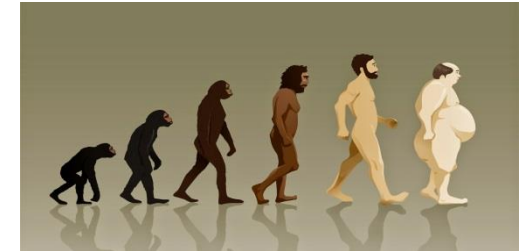
Деревья решений
Байесовские решения
Логистическая регрессия
Функции потенциалов
Ансамбли

ИИ-1

Формальные системы
Символьное программирование
Логическое программирование
Базы знаний
Экспертные системы
Онтологии
Семантические сети



Естественный интеллект



2023

Универсальные глубокие модели

2023

Универсальные модели, PE

2025

Универсальный гибридный искусственный интеллект

2030

Универсальный гибридный человеко-машинный интеллект

ПРИМЕНЕНИЕ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В АВИАЦИОННЫХ СИСТЕМАХ И КОМПЛЕКСАХ

Направления работ ГосНИИАС

Унифицированная платформа нейросетевой разработки «Платформа-ГНС» (2020)

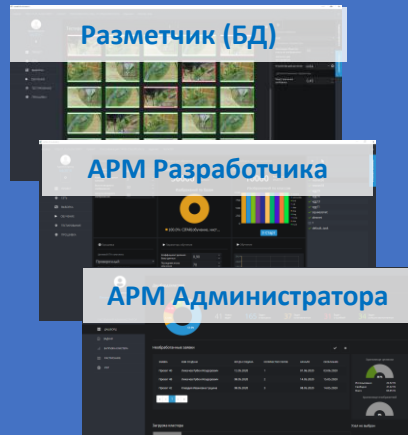
Интегрированная среда



Работа с данными

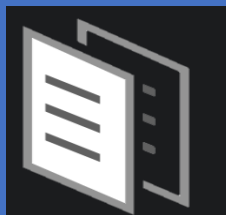
Формирование
и обучение ГНС

Аппаратная
реализация ГНС



Прикладные пакеты

- Готовые типовые решения
- Интегрированные решения
- Пользовательские решения



Типовые
задачи



Компьютерное
зрение



Анализ
сигналов

Средства
встраивания
решений
PlatformAPI



Унифицированная платформа

Сертифицируемый на НДВ исходный код

Импорт/экспорт, работа с фреймворками

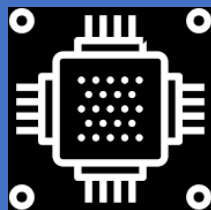
Готовые решения для типовых задач

Поддержка отечественного АО и ОС

Контроль доступа к данным и проектам

Низкие требования к квалификации ИТР

Средства аппаратной реализации



Библиотека машинного обучения (фреймворк)

PlatLib



Динамические
графы

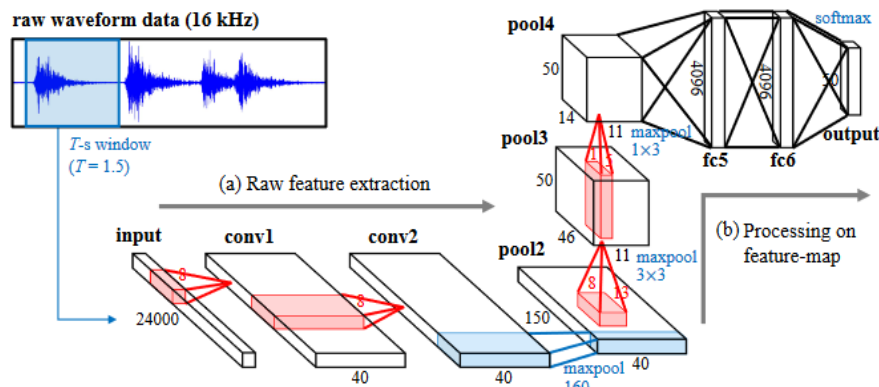
Распределенное
обучение

Унификация
с PyTorch

В ГосНИИАС создан и развивается полный отечественный программно-аппаратный стек технологий для ИИ

Проект «Автомат-ГНС» (2021-2023): новые функции и модули Платформы-ГНС

СПО «Сигнал-ГНС» - набор типовых решений для обработки одномерных и многомерных сигналов с целью решения задач классификации, регрессии, устранения шумов.



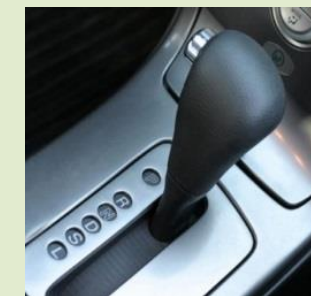
СПО «Авто-ГНС» — это набор программных модулей, которые позволяют пользователю в автоматическом режиме сформировать архитектуру ГНС (в задачах ТР) и подобрать оптимальные гиперпараметры обучения ГНС.



СПО «Аугмент-ГНС» — это набор программных модулей, которые используются в процессе обучения для определения оптимальной стратегии применения аугментаций (преобразований) к обучающей выборке.

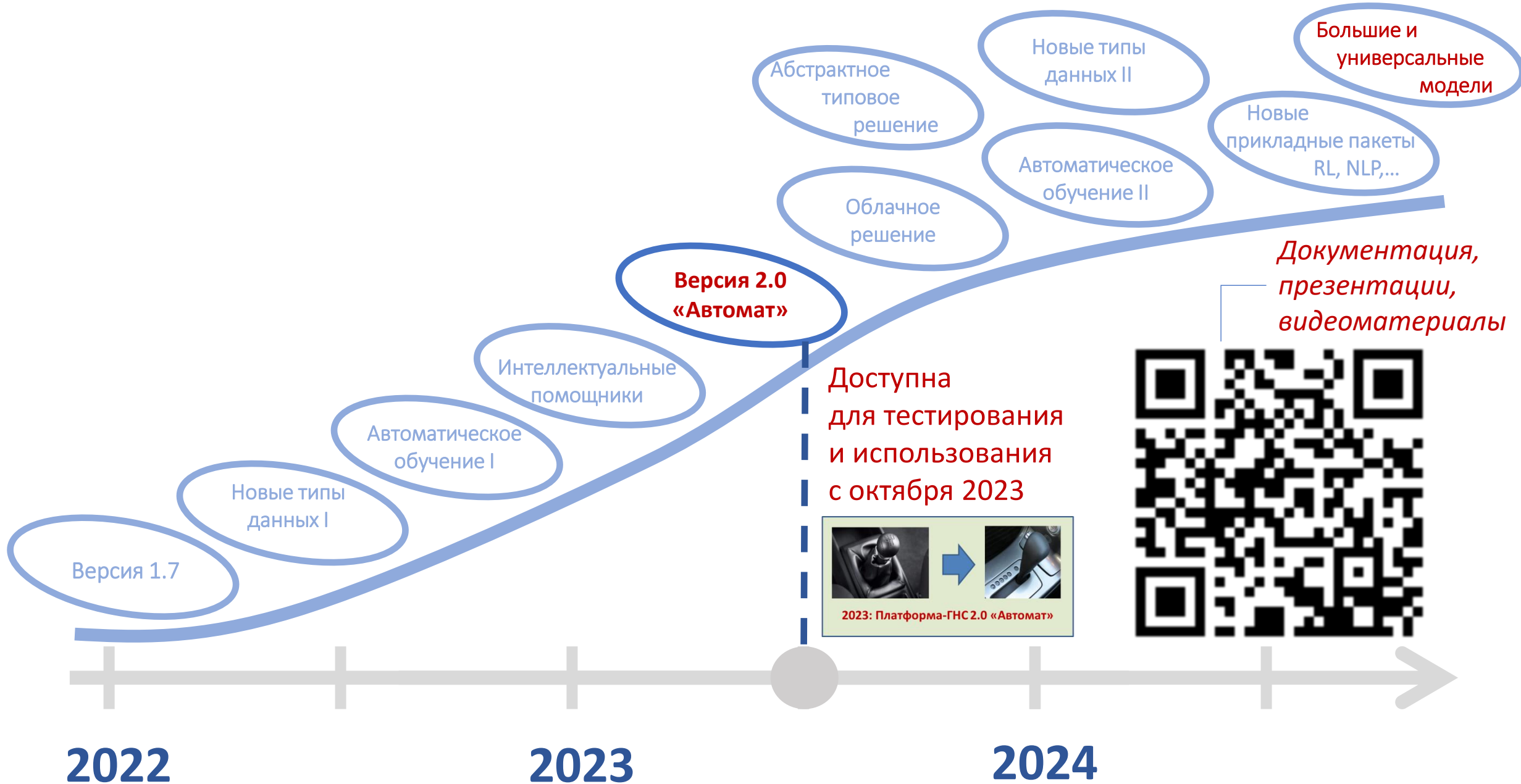
СПО «Советник-ГНС» — это набор программных модулей, которые анализируют действия пользователя, формируют и выдают пользователю ситуативные рекомендации на этапах подготовки данных, обучения и тестирования.

СПО «Синтез-ГНС» — это набор программных модулей, которые реализуют возможность реалистичного нейростевого синтеза изображений и сигналов.



2023: Платформа-ГНС 2.0 «Автомат»

Платформа-ГНС и PLAT: Перспективы развития (дорожная карта)



Платформа-ГНС и PLAT: Перспективы развития (уровни ИИ)

СТЗ, ML и ИИ для ЛА, БЛА, РТК...

Уровни ИИ

Развитие Платформы-ГНС

*Полная автоматизация
(замена человека
автономным ИИ,
советы или управление)*

Высокий уровень (Универсальный ИИ)
выполнение комплексных задач поиска, сбора, обработки
и анализа разнородных данных, управления сложными
объектами и системами, взаимодействия с людьми,
объяснения и анализа собственных действий

**Платформа-2024-26
(планируемое
развитие
Платформы)**

*СТЗ, частичная
автоматизация РТК
(выполнение отдельных
функций, которые ранее
выполнялись только
человеком)*

Средний уровень (Функциональный ИИ)
Работа с данными, семантическими сущностями
и символьными представлениями

**Платформа-2023
(версия 2.0)**

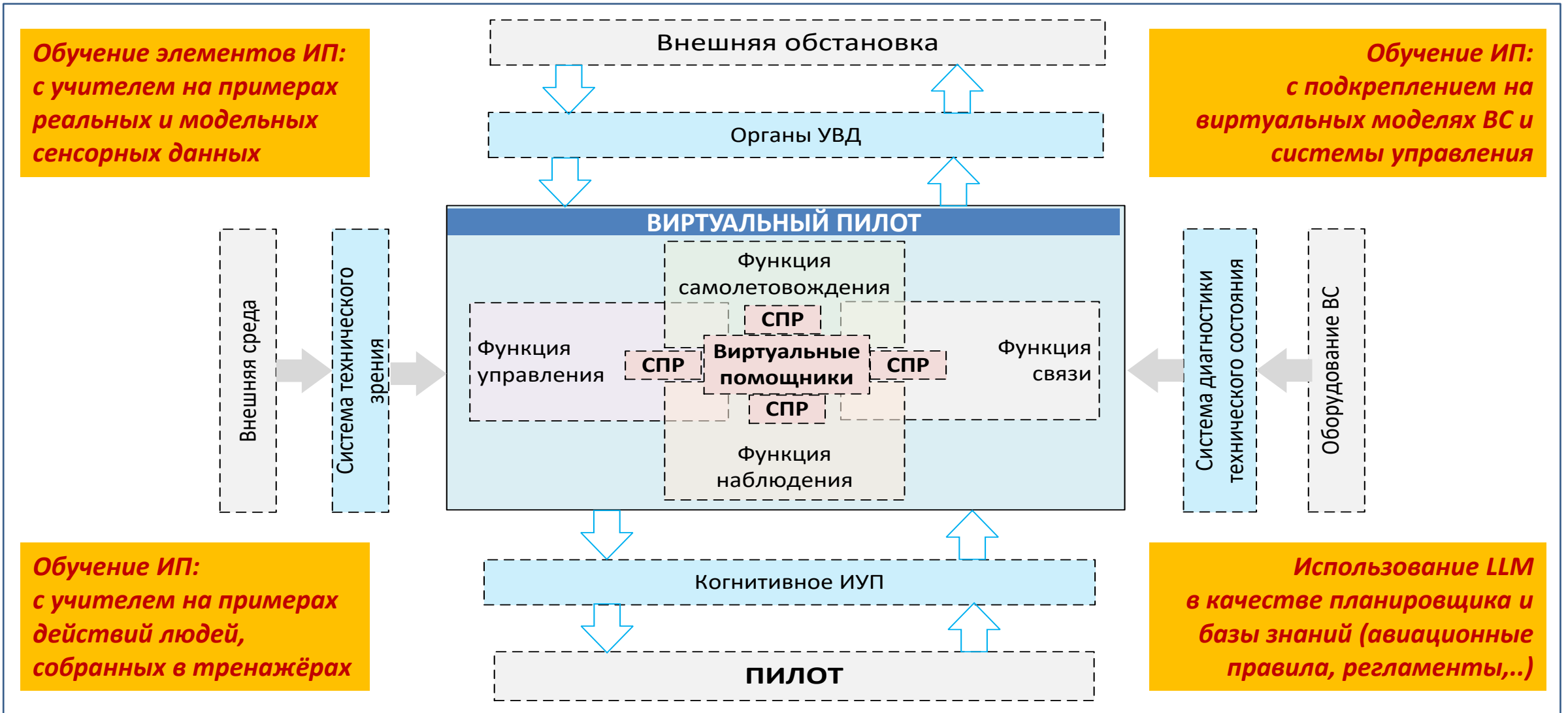
*Создание более
эффективных
адаптивных агрегатов*

Низкий уровень (Обучаемые машины и агрегаты)
Работа с данными и/или управление с использованием
машинного обучения, но без перехода на
семантический/символьный уровень

Платформа-ГНС как доверенная отечественная платформа ИИ – фундамент инструментария для внедрения ИИ в ВС ГА



Бортовые функции и интеллектуальные помощники



Интеллектуальное информационно-управляющее поле

CHECKLIST



ВОЗДУШНАЯ ОБСТАНОВКА



ЗЕМНОЙ РЕЛЬЕФ



ПОСАДКА



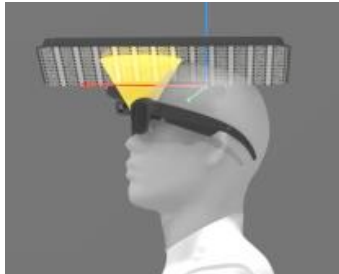
РУЛЕНИЕ



СИГНАЛИЗАЦИЯ



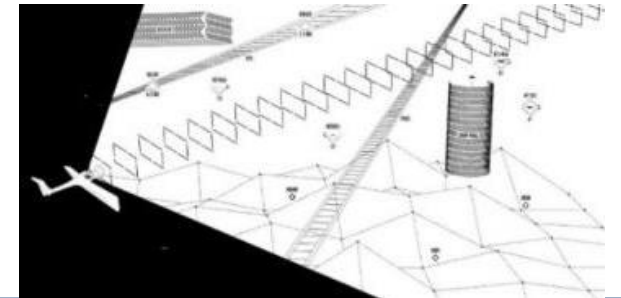
ОПРЕДЕЛЕНИЕ ПОЛОЖЕНИЯ ГОЛОВЫ



AR-очки



ЦИФРОВАЯ МОДЕЛЬ ВП

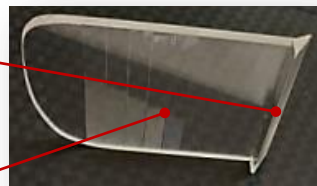


Проекционная система Full HD

КОНСТРУКЦИЯ



Призма



Отражающие грани

МУЛЬТИМОДАЛЬНОЕ УПРАВЛЕНИЕ



ЦИФРОВАЯ МОДЕЛЬ КАБИНЫ



Демонстрация логики работы виртуальных помощников

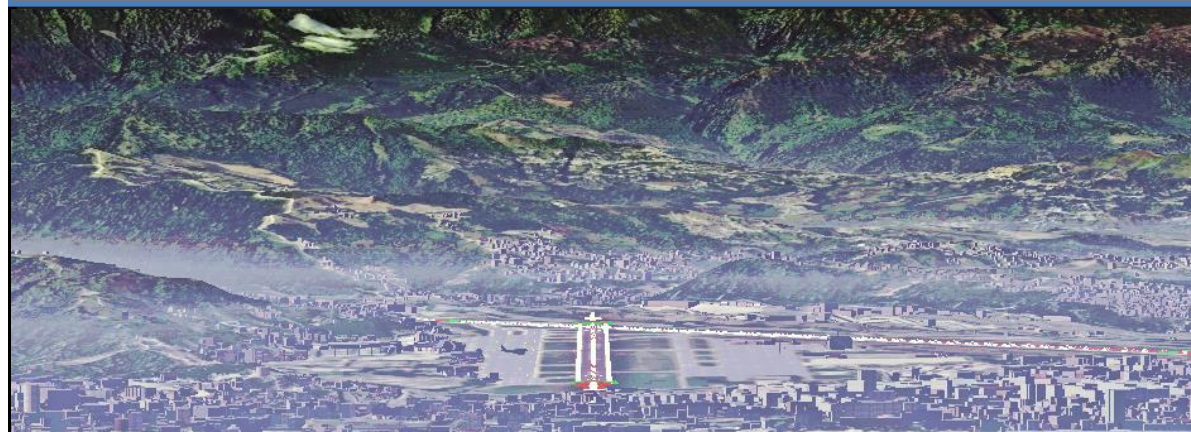


Сенсорная информация СТЗ: ТВ канал

Pos:	IMU	Выч.	Ошибка
X (м):	-2235.00	-2254.10	19.10
Y (м):	127.84	136.29	-8.45
Z (м):	-2.45	-1.96	-0.49
Yaw:	-0.61	-0.63	
Pitch:	0.60	0.41	
Roll:	-0.20	-0.02	



Автоматическая посадка по данным СТЗ



Система улучшенного видения ТВ канала

Диспетчер 17:10:06
Посадка 220, до зоны приземления.
Командир ВС 17:10:07
Малый газ.

Система 17:10:08
Включен режим улучшенного видения.

Командир ВС 17:10:09
Реверс включен.
Командир ВС 17:10:10
Спойлеры, щитки выпущены.
Командир ВС 17:10:11
Торможу.

Система 17:10:12
Обнаружено начало ВПП.
Система 17:10:13
Координаты ВПП получены.

Командир ВС 17:10:14
Реверс выключить.

Интеллектуальный борт как «чат» ИИ-агентов

Интеллектуальные технологии для авиационных систем улучшенного, синтезированного и комбинированного видения

Создан прототип СУВ на базе платформы ИМА

**ОПТИКО-ЭЛЕКТРОННЫЙ БЛОК СУВ И ВЫЧИСЛИТЕЛЬНЫЙ
МОДУЛЬ ИЗ СОСТАВА ПЛАТФОРМЫ ИМА**



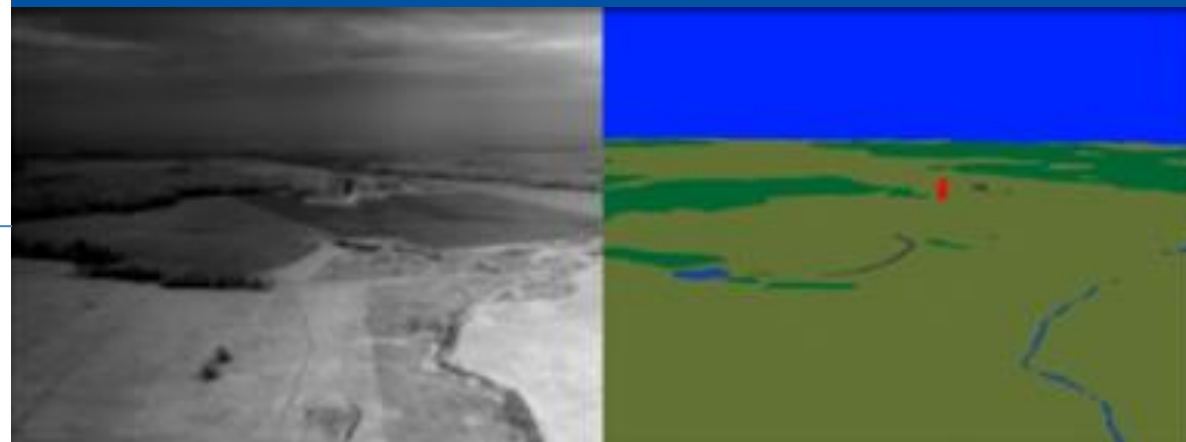
Элементы ИИ: CNN и GAN для комплексирования и сегментации изображений, а также оригинальные методики машинного обучения, учитывающие оценку качества изображений типовых сцен экспертами-пилотами.

Развитие ПО СУВ на основе нейросетевых подходов

КОМПЛЕКСИРОВАНИЕ ИЗОБРАЖЕНИЙ



СЕМАНТИЧЕСКАЯ СЕГМЕНТАЦИЯ СЦЕНЫ



Роль и функции системы внешнего видения (СТЗ) в общей схеме интеллектуализации перспективного борта ВС

Навигация

Новые свойства:

- **Определение относительного и абсолютного положения ВС (оптико-навигационный контур коррекции)**
- Определение отказа навигационного компонента на посадке за счет функциональной избыточности

Наблюдение

Новые свойства:

- **Распознавание элементов ВПП и РД**
- **Распознавание символьных элементов ВПП**
- **Распознавание разметки ВПП и РД**
- **Обнаружение и распознавание препятствий на ВПП и РД**
- **Классификация ВС**
- **Расчет удаления до ВС**
- Определение конфликтной ситуации
- Приоритизация конфликтной ситуации

Интеллектуальные функции СТЗ

СТЗ = Датчики + Нейровычислитель

Управление

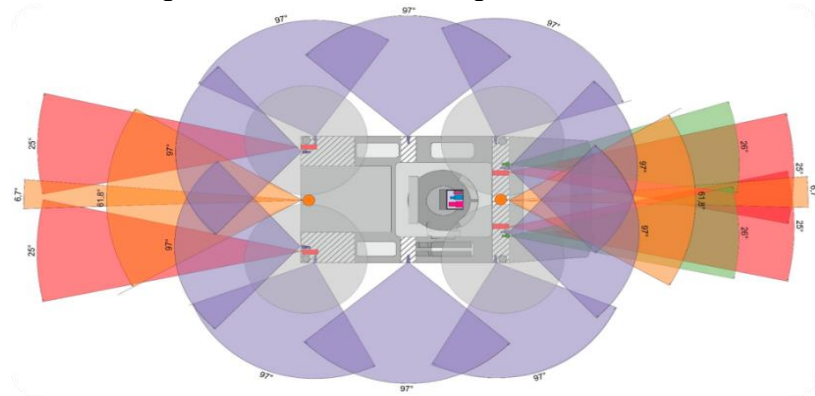
Новые свойства:

- **Автоматическая посадка**, уход на второй круг и выход в зону ожидания
- **Автоматический пробег**
- **Автоматическое руление, интеллектуальный круиз-контроль**
- Взлёт, прерванный взлёт
- Ремаршрутизация в сложные метеоусловиях при отказах систем ВС
- Прогноз параметров полета на базе технологий ИИ с учетом текущего состояния ВС
- Оптимизация параметров полета с учетом факторов внешней обстановки
- Автоматический возврат на аэродром посадки

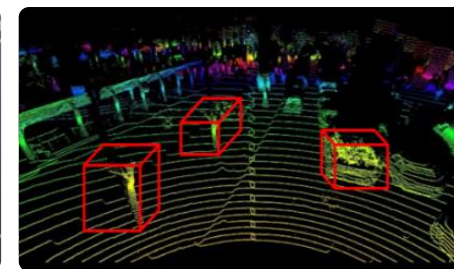
Интеллектуальные функции СТЗ

Технологии ГосНИИАС в области систем автоматического вождения как ИТЗ для системы автоматического руления ВС (опыт РТП Маркер)

Поля зрения датчиков сенсорной системы РТП



- Построение трехмерной модели местности
- Интеграция в общую информационную модель



- Поиск и распознавание препятствий
- Построение карты проходимости

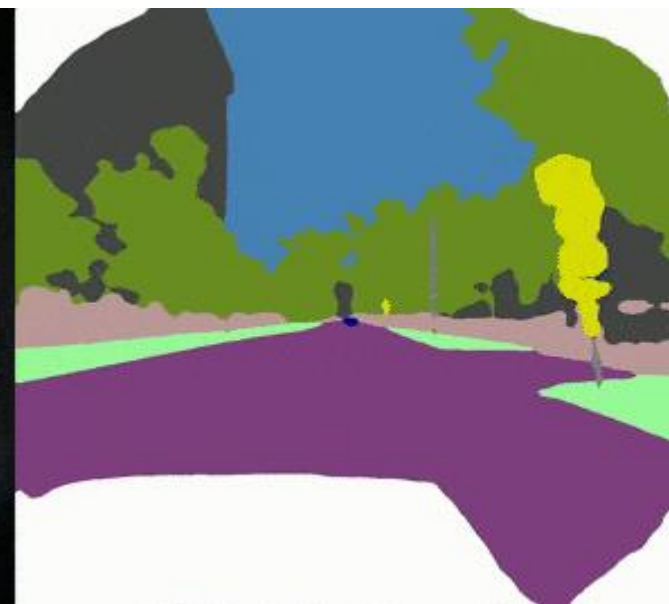


- Семантическая сегментация сцены

- Макет УМПН (поворотная платформа + ТВ-камера + дальномер)
- Длиннофокусные камеры x2
- PTZ-камера x2
- Лидары x4
- Коммуникационный шкаф
- Короткофокусные камеры x6



Мобильная лаборатория ГосНИИАС для отработки режимов автоматического руления



Пример работы алгоритмов семантической сегментации

От слабого ИИ к общему универсальному интеллекту (обзор тенденций 2020-2023)

Визильтер Юрий Валентинович, д.ф.-м.н., проф. РАН, директор по направлению – руководитель научного комплекса «Искусственный интеллект и техническое зрение» ФАУ «ГосНИИАС», viz@gosniias.ru

Спасибо за внимание!

Научно-исследовательский семинар
РАИИ и ФИЦ ИУ РАН
«Проблемы искусственного интеллекта»
Москва, ФИЦ ИУ РАН, 31.01.2024

