THE TRADER OF TOMORROW: REVISITED

Dr. John Ashley, November 2021

The Trader of the Future

Refresher

Augmented and Artificial Intelligence

NLP

The New Stuff

GPT-3 class models

Sidebar: Petaflops!

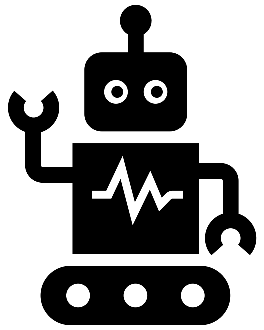
Synthetic Data Experiments

Prompting; Wasserstein Distance;

Datasets & Some Results – Work by Yi Dong, Manny Scoullos

AI FOR TRADING

Selected Use Cases



Augmented Intelligence for Discretionary Traders

NLP

- Text Prioritization
- Text Summarization
- Named Entity Recognition & Knowledge Graphs

Artificial Intelligence for Algo Traders

Algo Development

- Time Series via RNN / Temporal CNN
- **Synthetic Data / VAE & GAN (backtesting)**

Sentiment Analysis – News, Social Media, Regulatory Filings

"alt data"

Optimal execution (Reinforcement Learning)

Deep Learning for Pricing and Risk

AI FOR TRADING

Selected Use Cases

Augmented Intelligence for Discretionary Traders

NLP

- Text Prioritization
- Text Summarization
- Named Entity Recognition & Knowledge Graphs

Connected?

Artificial Intelligence for Algo Traders

Algo Development

- Time Series via RNN / Temporal CNN
- **Synthetic Data / VAE & GAN (backtesting)**

Sentiment Analysis – News, Social Media, Regulatory Filings

"alt data"
Optimal execution (Reinforcement Learning)
Deep Learning for Pricing and Risk

LANGUAGE UNDERSTANDING IMPROVEMENT

Reaching human level

GLUE Aggregate Score

Detect grammatical errors

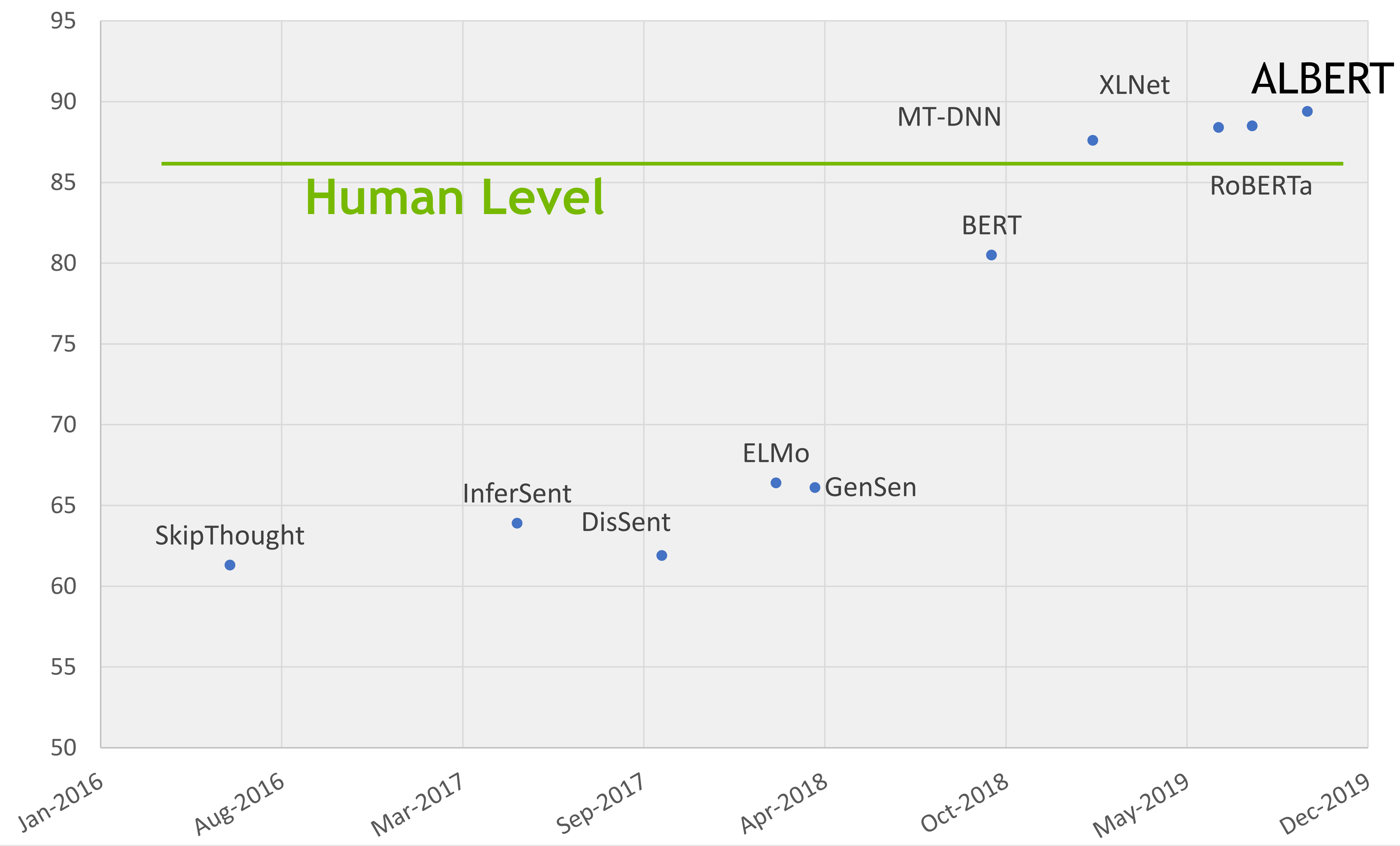
Predict if movie review is positive or negative

Decide if an abstract correctly summarizes an article

Sentence-level Semantic equivalence

Basic reading comprehension

Pronoun disambiguation



https://gluebenchmark.com/

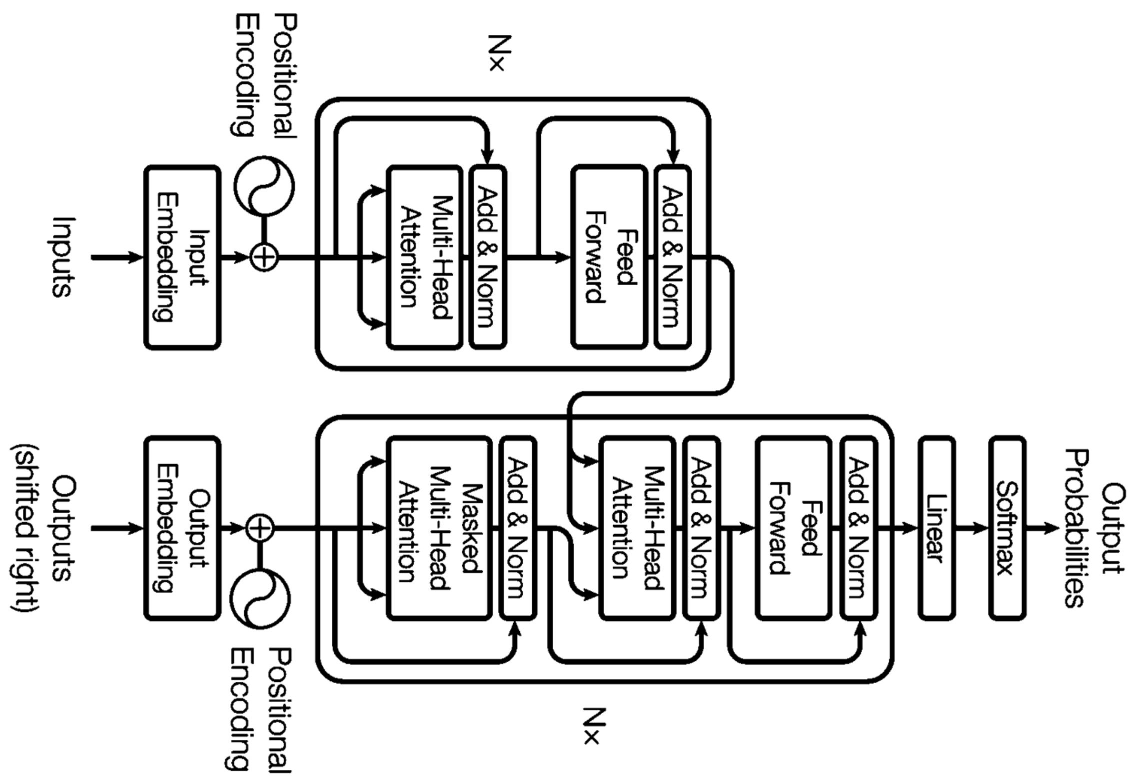
NATURAL LANGUAGE UNDERSTANDING

BERT universal language model

Input: Two sentences with 15% of words masked out

1 = “Initially he supported himself and his [redacted] by farming on a plot [redacted] family land.”

2 = “[redacted] in turn attracted the attention of [redacted] *St.* [redacted] *Post-Dispatch*, which sent a reporter to Murray to [redacted] review Stubblefield's wireless [redacted].”



Output 1: Reconstruct missing words

family, of
this, the, Louis, personally,
telephone

Output 2: Is two the next sentence after one?

NOT_NEXT_SENTENCE

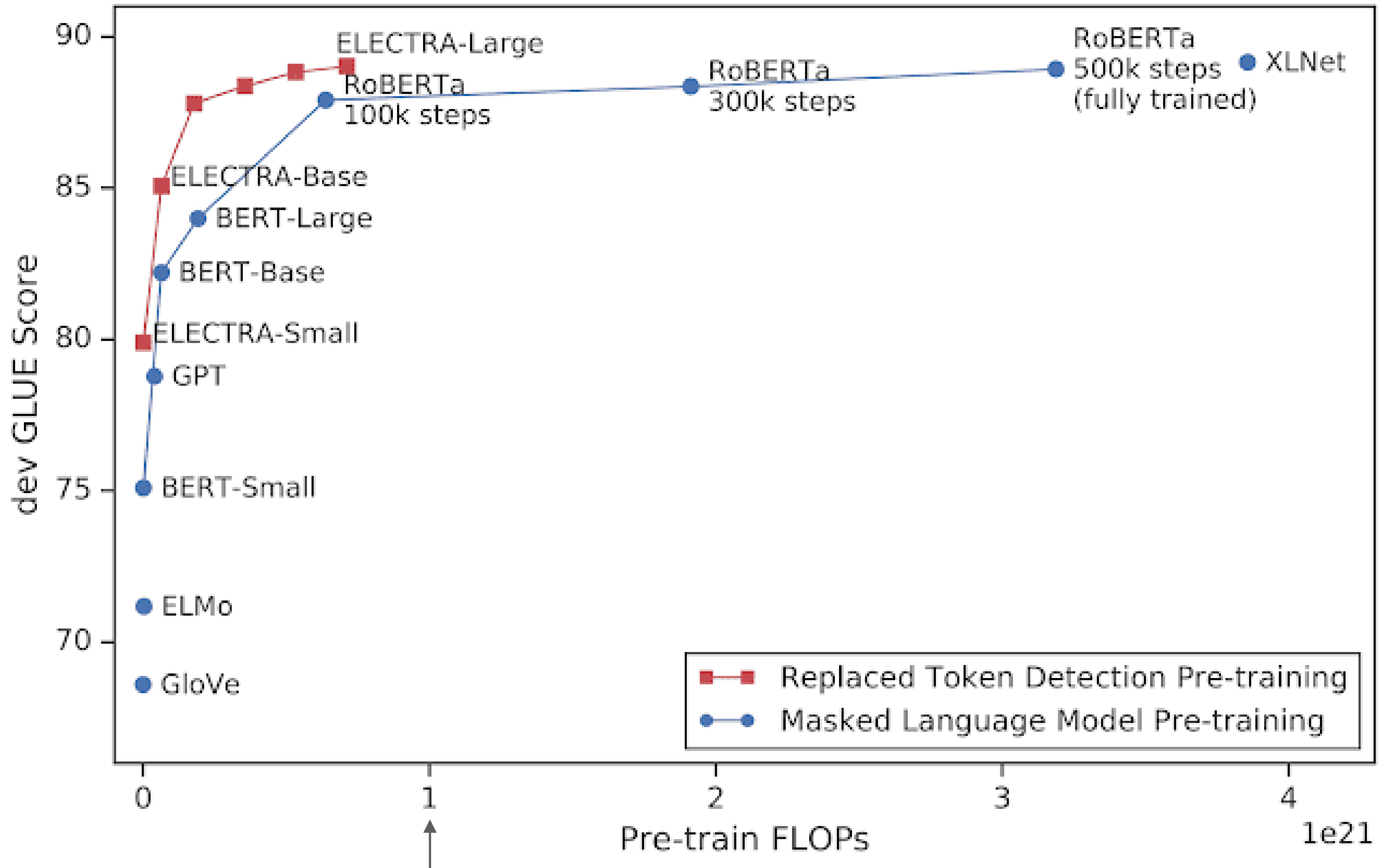
https://arxiv.org/abs/1810.04805

21x parameter growth

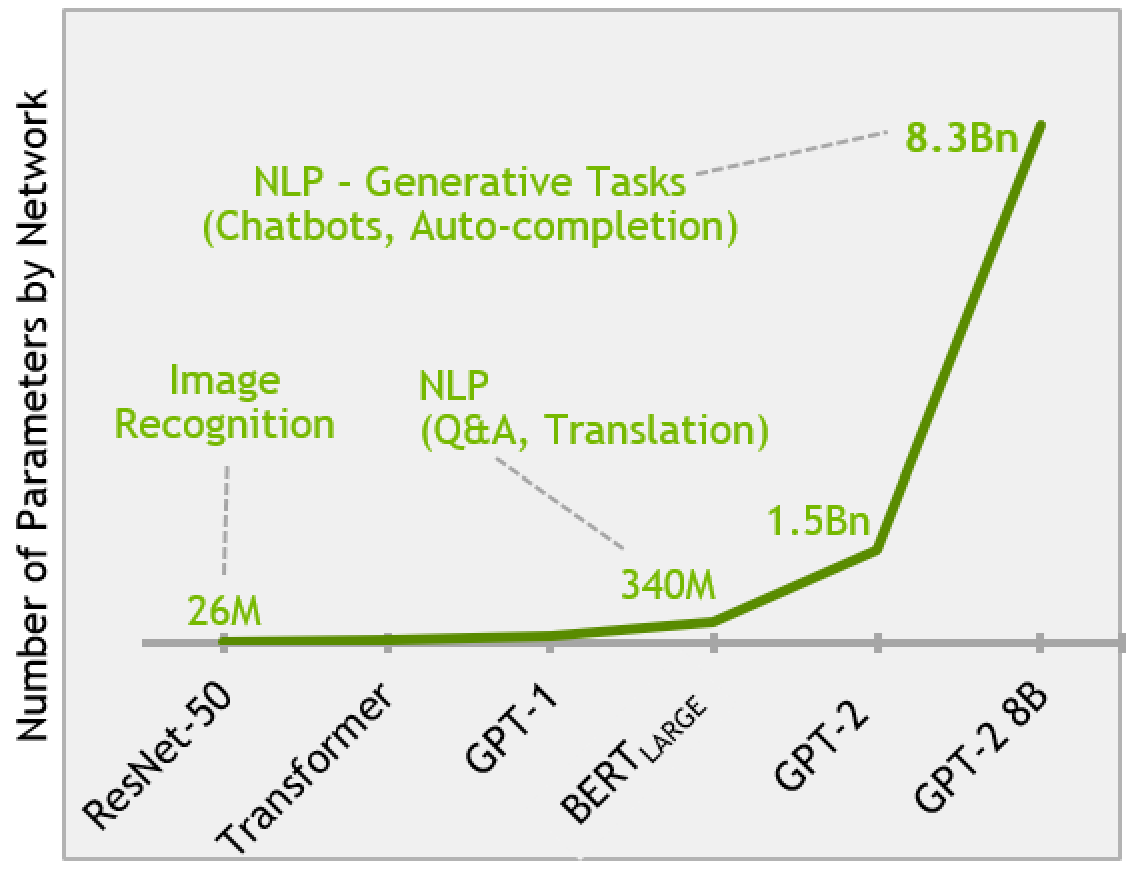
NLP MODELS ARE LARGE

The Training and Inference cost is high

GPT-3 = 175 Billion Parameters!



1 Zettaflop = 1,000 Exaflops

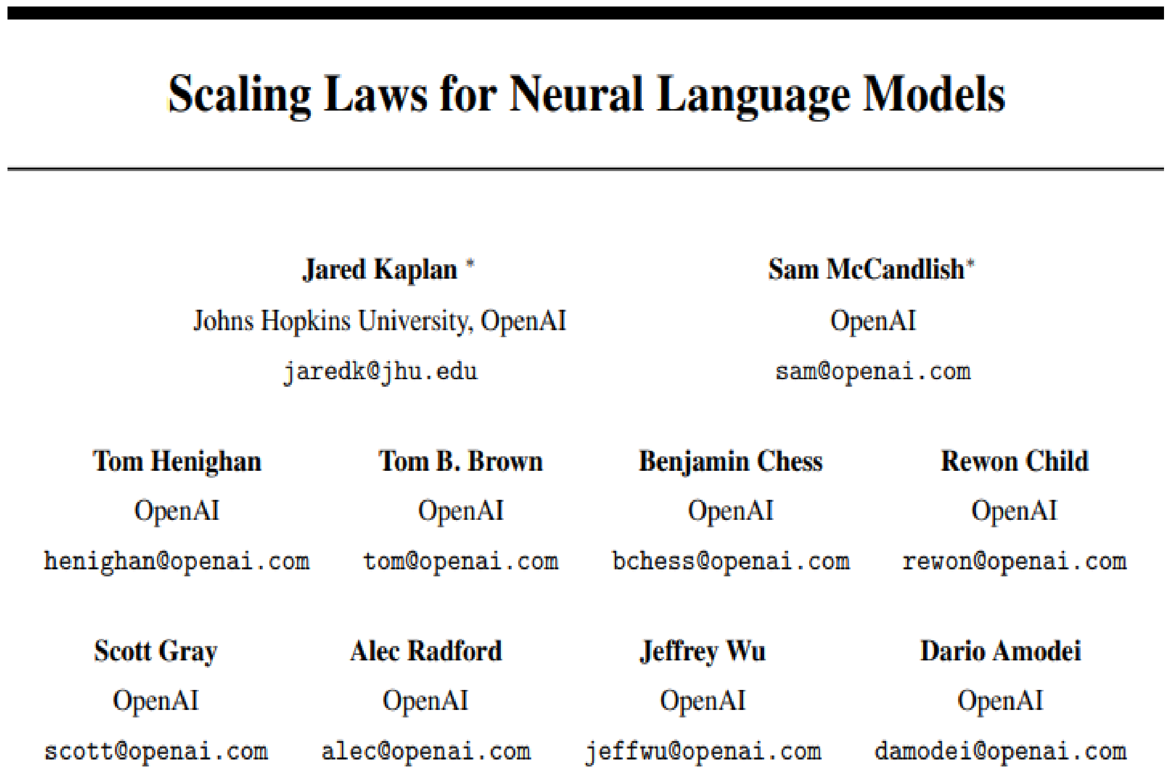


EXPLODING MODEL SIZE

Complexity to Train

https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/

WHY LARGE MODELS?



Scaling Laws for Neural Language Models

Jared Kaplan *
Johns Hopkins University, OpenAI
jaredk@jhu.edu

Sam McCandlish*
OpenAI
sam@openai.com

Tom Henighan
OpenAI
henighan@openai.com

Tom B. Brown
OpenAI
tom@openai.com

Benjamin Chess
OpenAI
bchess@openai.com

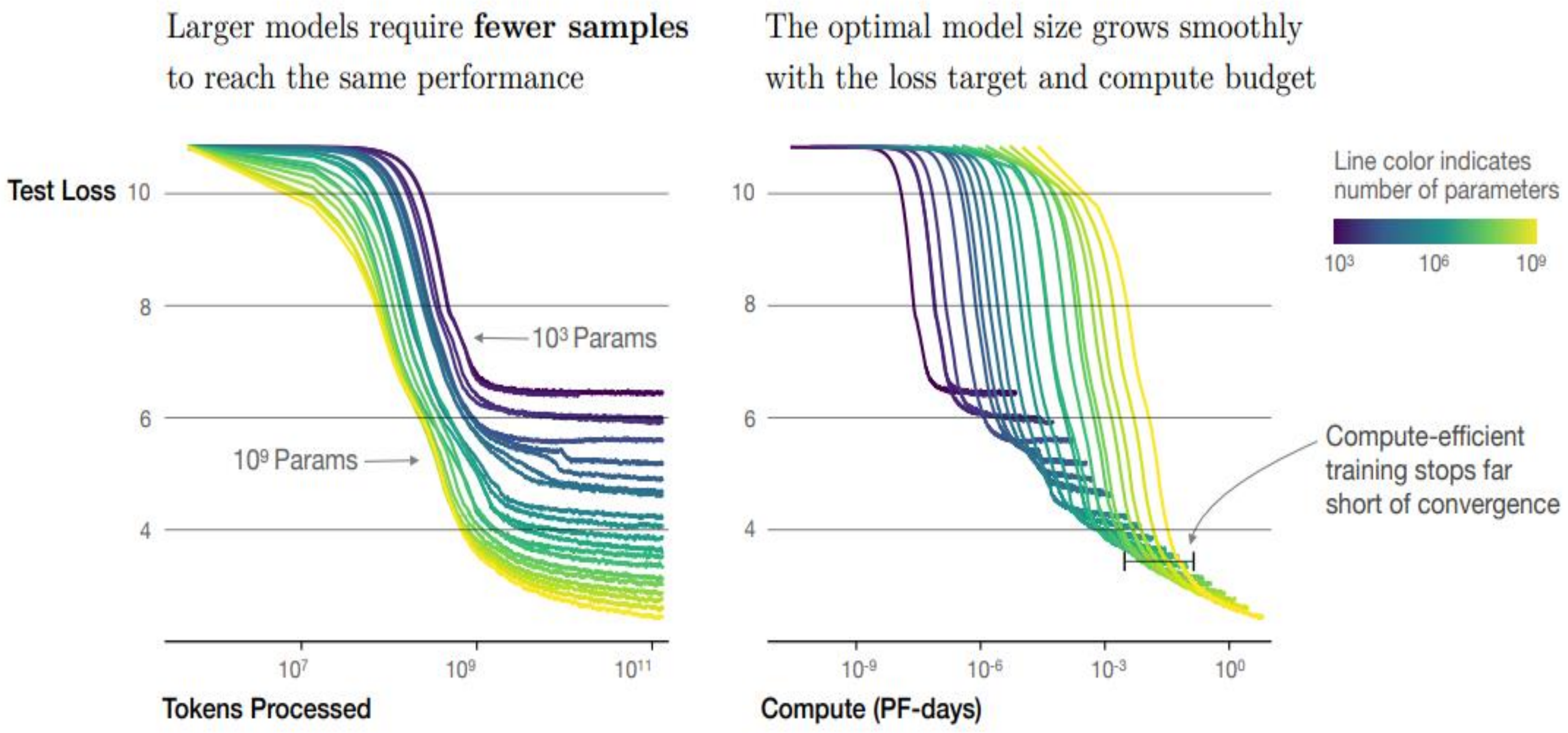
Rewon Child
OpenAI
rewon@openai.com

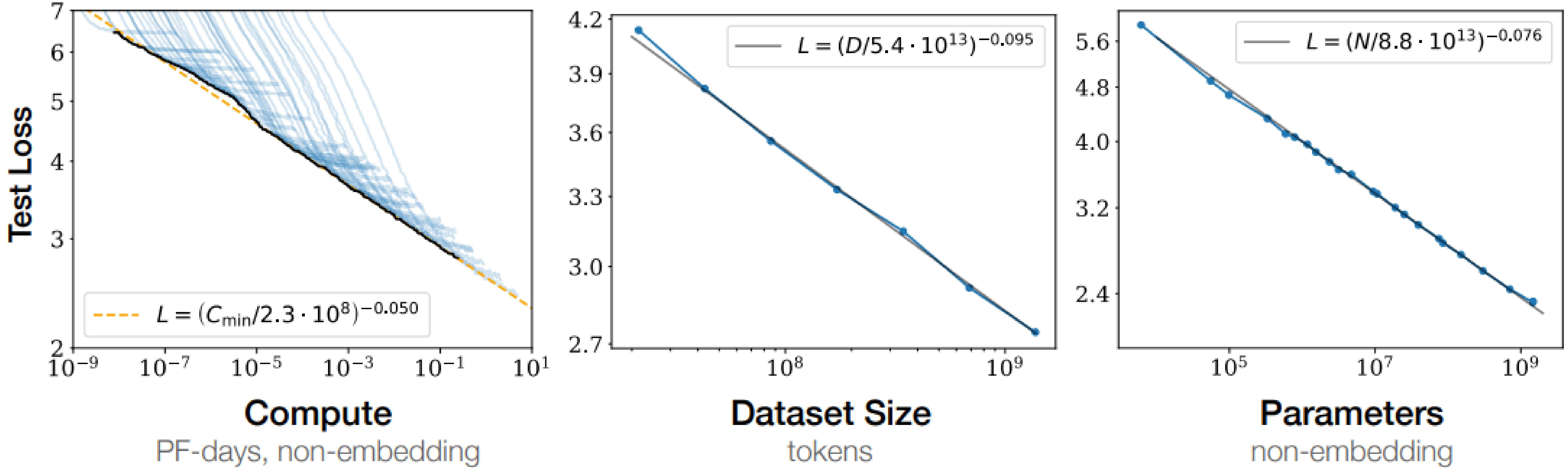
Scott Gray
OpenAI
scott@openai.com

Alec Radford
OpenAI
alec@openai.com

Jeffrey Wu
OpenAI
jeffwu@openai.com

Dario Amodei
OpenAI
damodei@openai.com





SIDEBAR: HOW MUCH COMPUTE IS A PETAFLOP?

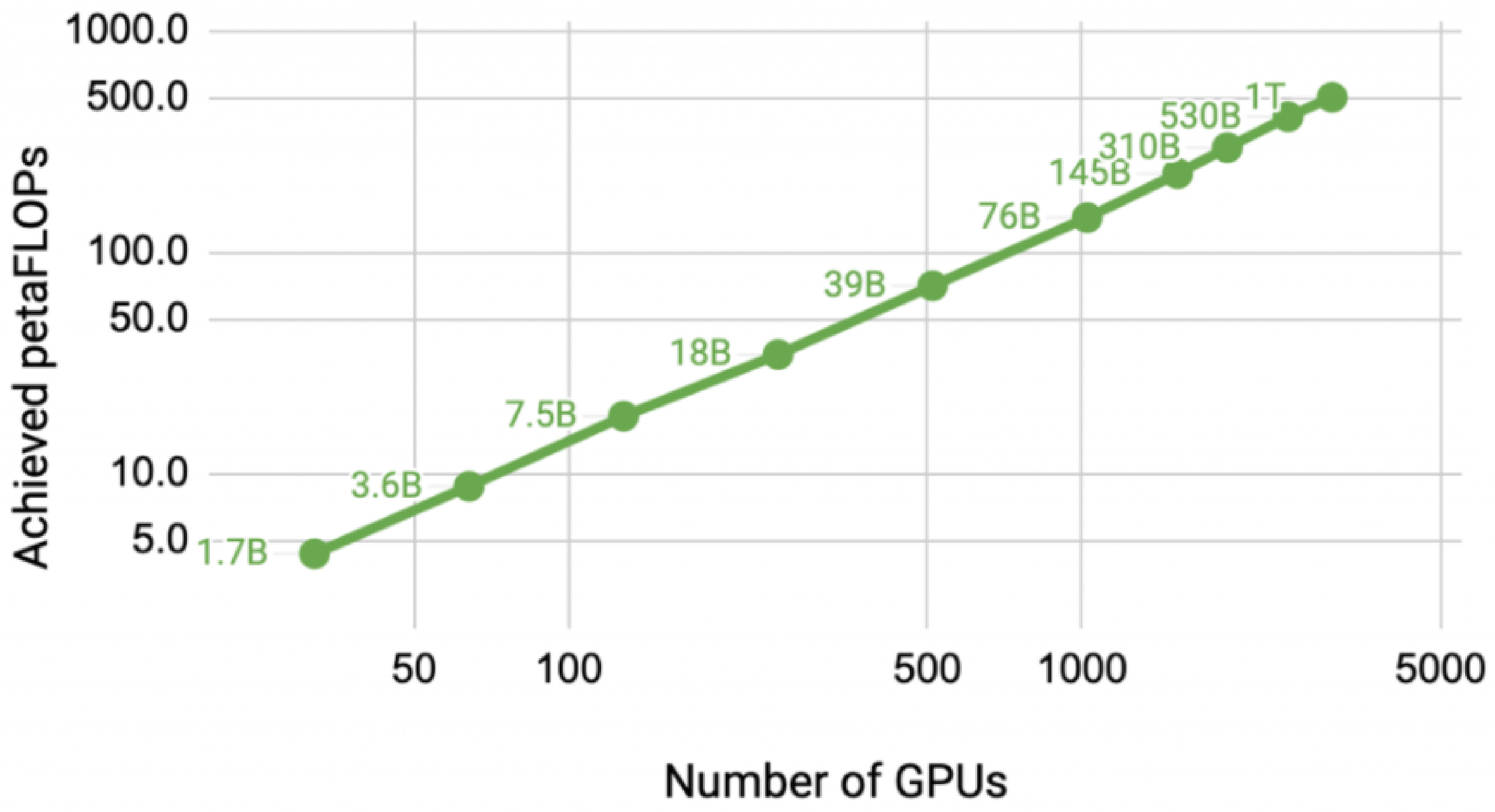
This is one reason why we built DGX A100 640GB!

Scaling Language Model Training to a Trillion Parameters Using Megatron

By Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand and Bryan Catanzaro

Discuss (1) Share 0 Like

Tags: Conversational AI / NLP, DGX A100, Megatron, model parallelism, pipeline parallelism



STAC-A2™ (beta 2) Report Card

STAC-A2 Pack for CUDA (Rev G) / 8 x NVIDIA A100 SXM4 80GiB / 2TiB DRAM / NVIDIA DGX A100 / OpenShift 4.8.3 (RHCOS 48.84)

(SUT ID: NVDA210914)

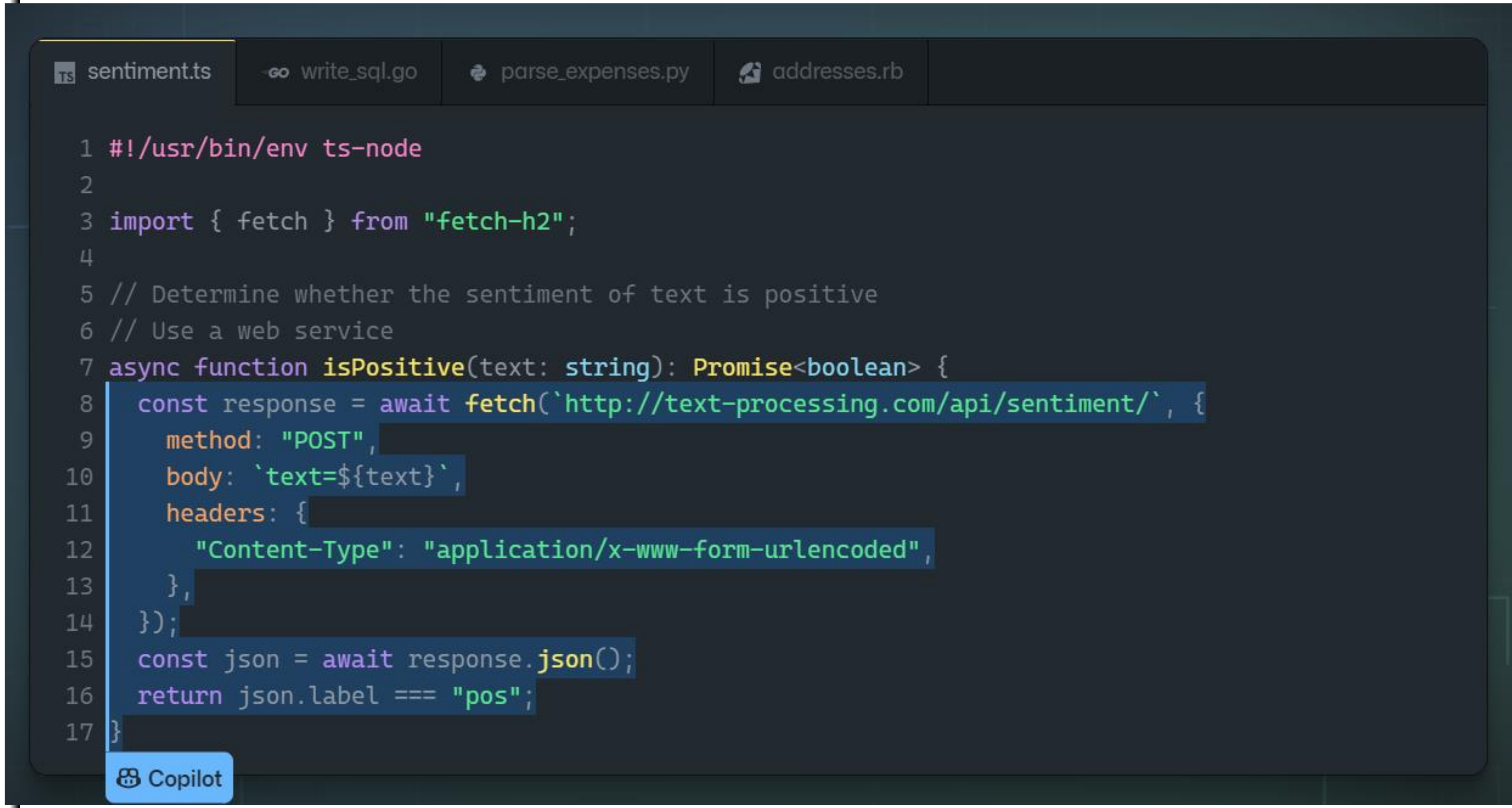
STAC-A2.β2.HPORTFOLIO.SPEED	Ratio of options completed to elapsed time	357.1	options per second
STAC-A2.β2.HPORTFOLIO.ENERG_EFF	Energy efficiency = HPORTFOLIO.OPTIONS_DONE / Energy Consumed	280,607	options per kWh
STAC-A2.β2.HPORTFOLIO.SPACE_EFF	Space efficiency = HPORTFOLIO.SPEED / Effective Volume	100.1	options per hour per cubic inch
STAC-A2.β2.GREEKS.TIME	Seconds to compute all Greeks with 5 assets, 25K paths, and 252 timesteps.*	WARM	0.012
		COLD	0.398
STAC-A2.β2.GREEKS.10-100k-1260.TIME	Seconds to compute all Greeks with 10 assets, 100K paths, and 1260 timesteps.*	WARM	0.7
		COLD	2.6
STAC-A2.β2.GREEKS.MAX_ASSETS	Max assets completed in 10 minutes with 25K paths and 252 timesteps (using cold test runs).		340
STAC-A2.β2.GREEKS.MAX_PATHS	Max paths completed in 10 minutes with 5 assets and 252 timesteps (using cold test runs).		204,800,000

DGX A100 640 GB Peak Flops
FP64 : 77 TF
FP32 : 156 TF
TF32 : 1,248 TF

“ANY SUFFICIENTLY ADVANCED TECHNOLOGY IS INDISTINGUISHABLE FROM MAGIC.”

– ARTHUR C. CLARKE

THE MAGIC OF GPT3
CREATING CONTEXT VIA PROMPTING



EXPERIMENT ORDER DATA

```
108 sec19 add buy 0.329 0.0 121
841 sec22 add sell 0.674 0.0 82
0 sec22 delete sell 0.674 0.0 82
0 sec7 add buy 0.51 0.001 123
517 sec17 delete sell 0.194 0.0 123
0 sec8 add sell 0.512 0.0 30
0 sec4 modify sell 0.449 0.002 30
0 sec9 modify sell 0.255 0.002 30
0 sec18 modify buy 0.704 0.008 30
0 sec21 modify buy 0.182 0.006 30
0 sec21 modify sell 0.184 0.005 30
0 sec21 modify buy 0.184 0.005 30
0 sec12 modify sell 0.561 0.001 30
0 sec6 modify buy 0.35 0.001 30
0 sec17 modify sell 0.25 0.001 30
0 sec17 modify sell 0.25 0.001 30
0 sec3 modify sell 0.524 0.001 30
0 sec0 add sell 0.984 0.0 12
0 sec8 modify buy 0.508 0.001 12
851 sec18 modify buy 0.705 0.008 12
0 sec18 modify buy 0.705 0.008 12
0 sec12 modify sell 0.527 0.001 12
0 sec9 modify sell 0.218 0.005 12
0 sec17 modify buy 0.188 0.001 12
0 sec7 modify buy 0.477 0.001 12
0 sec6 modify buy 0.314 0.001 12
0 sec9 delete sell 0.218 0.005 30
0 sec9 add sell 0.261 0.005 84
299 sec8 modify buy 0.508 0.0 84
911 sec0 delete buy 0.984 0.0 84
19 sec0 add buy 0.984 0.0 103
720 sec20 modify buy 0.519 0.0 103
633 sec20 modify buy 0.519 0.0 103
512 sec20 delete sell 0.519 0.0 103
315 sec0 add buy 0.984 0.0 26
19 sec0 add buy 0.984 0.0 115
```

Convert it to NLP problem

{"text": "0 sec17 add buy 0.2 0.002 84\n0 sec17 delete buy 0.169 0.001 84\n0 sec17 add buy 0.194 0.001 94\n0 sec17…

{"text": "0 sec17 add buy 0.2 0.002 8\n0 sec17 delete buy 0.169 0.001 8\n0 sec17 add buy 0.194 0.001 102\n0 sec17…

EXPERIMENT CREDIT CARD DATA

Convert it to NLP problem

```
user,card,date,year,month,day,time,hour,minute,amount,use chip,merchant name,merchant city,merchant state,zip,mcc,errors,is_fraud
791,1,1991-01-02 07:10:00,1991,1,2,07:10,7,10,68.0,Swipe Transaction,2027553650310142703,Burke,VA,22015,5541,,0
791,1,1991-01-02 07:17:00,1991,1,2,07:17,7,17,-68.0,Swipe Transaction,2027553650310142703,Burke,VA,22015,5541,,0
791,1,1991-01-02 07:21:00,1991,1,2,07:21,7,21,113.62,Swipe Transaction,2027553650310142703,Burke,VA,22015,5541,,0
791,1,1991-01-02 17:30:00,1991,1,2,17:30,17,30,114.73,Swipe Transaction,-7269691894846892021,Burke,VA,22015,5411,,0
791,1,1991-01-03 09:03:00,1991,1,3,09:03,9,3,251.71,Swipe Transaction,-3693650930986299431,Burke,VA,22015,4814,,0
791,1,1991-01-03 11:14:00,1991,1,3,11:14,11,14,16.28,Swipe Transaction,-7269691894846892021,Burke,VA,22015,5411,,0
791,1,1991-01-03 12:46:00,1991,1,3,12:46,12,46,172.0,Swipe Transaction,3189517333335617109,Fairfax,VA,22030,5311,,0
791,1,1991-01-04 11:09:00,1991,1,4,11:09,11,9,16.63,Swipe Transaction,5701841789931834110,Burke,VA,22015,5411,,0
791,1,1991-01-04 13:56:00,1991,1,4,13:56,13,56,27.0,Swipe Transaction,-8194579483471190227,Burke,VA,22015,5211,,0
```

encode

```
1690 1 21 8 16 11 50 19.45 0 11607 1773 10 0
1690 1 21 8 16 16 43 8.18 0 12 1773 11 0
1690 0 21 8 17 8 36 3.3 0 54 1773 1 0
1690 0 21 8 17 9 35 53.06 0 54 1773 1 0
1690 1 21 8 17 11 34 22.21 0 12728 1773 1 0
1690 1 21 8 18 0 11 130.05 0 18 1793 14 0
1690 1 21 8 18 1 40 49.3 0 253 18801 8 0
1690 0 21 8 18 11 42 17.54 0 30 589 23 0
1690 1 21 8 18 19 17 28.35 0 54 1773 1 0
```

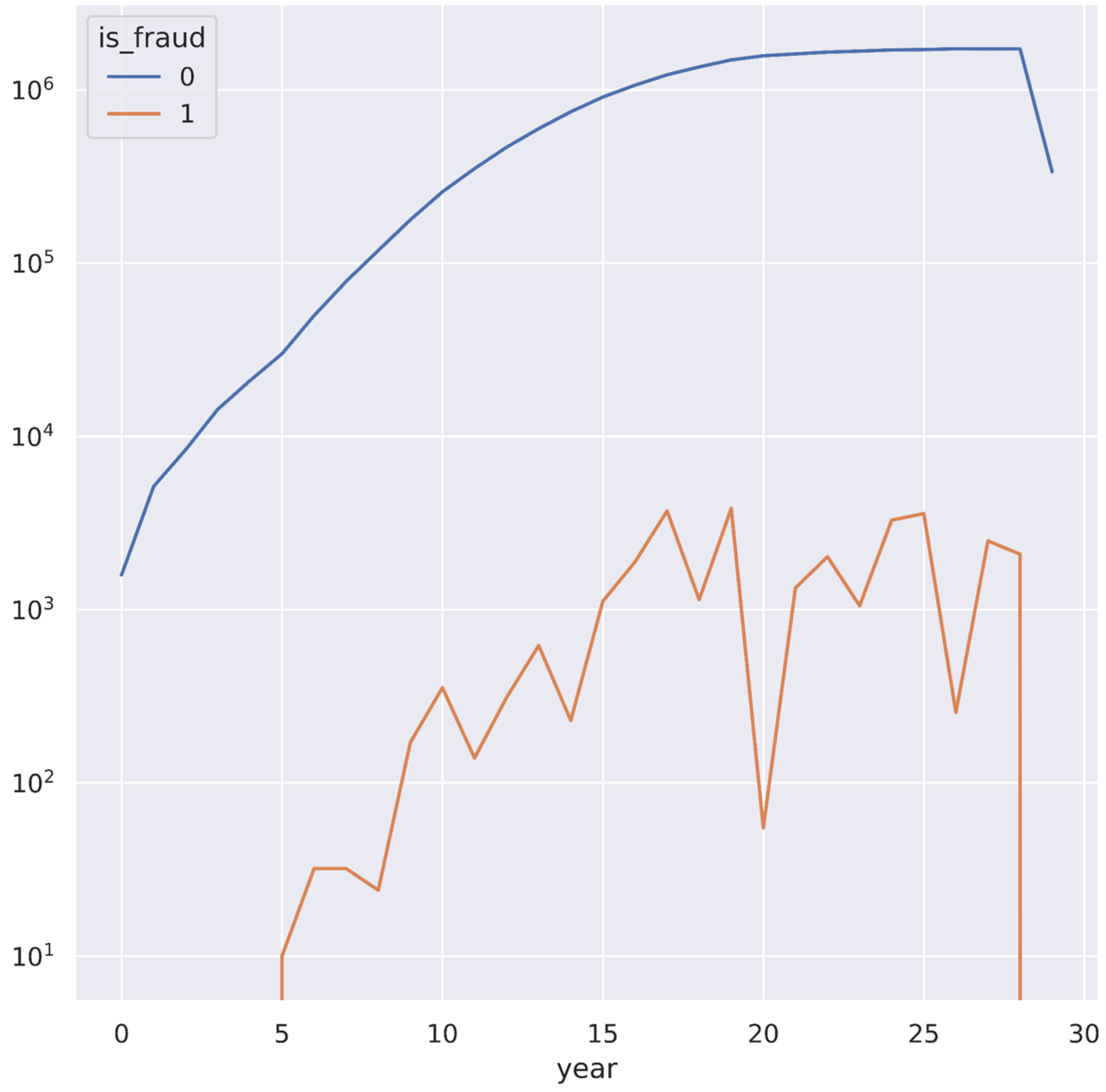
MOTIVATION - IBM TABFORMER DATASET

- Realistic rule generated synthetic dataset for payments fraud
- Favorable usage license
- 24M transactions, 2000 users, 100K merchants across 30 years (1990-2020)

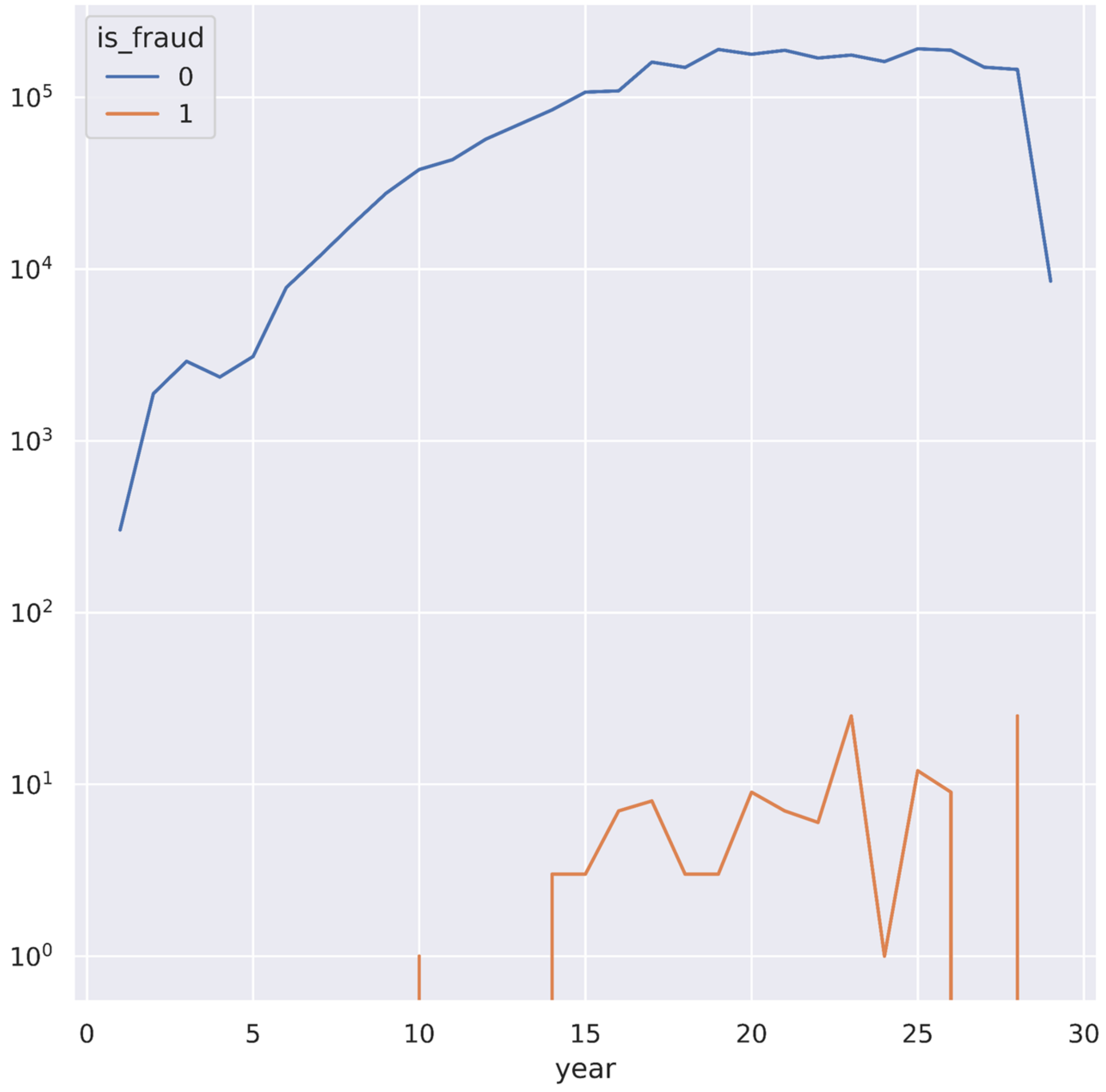
https://github.com/IBM/TabFormer

DOES IT WORK? FRAUD VS NOT

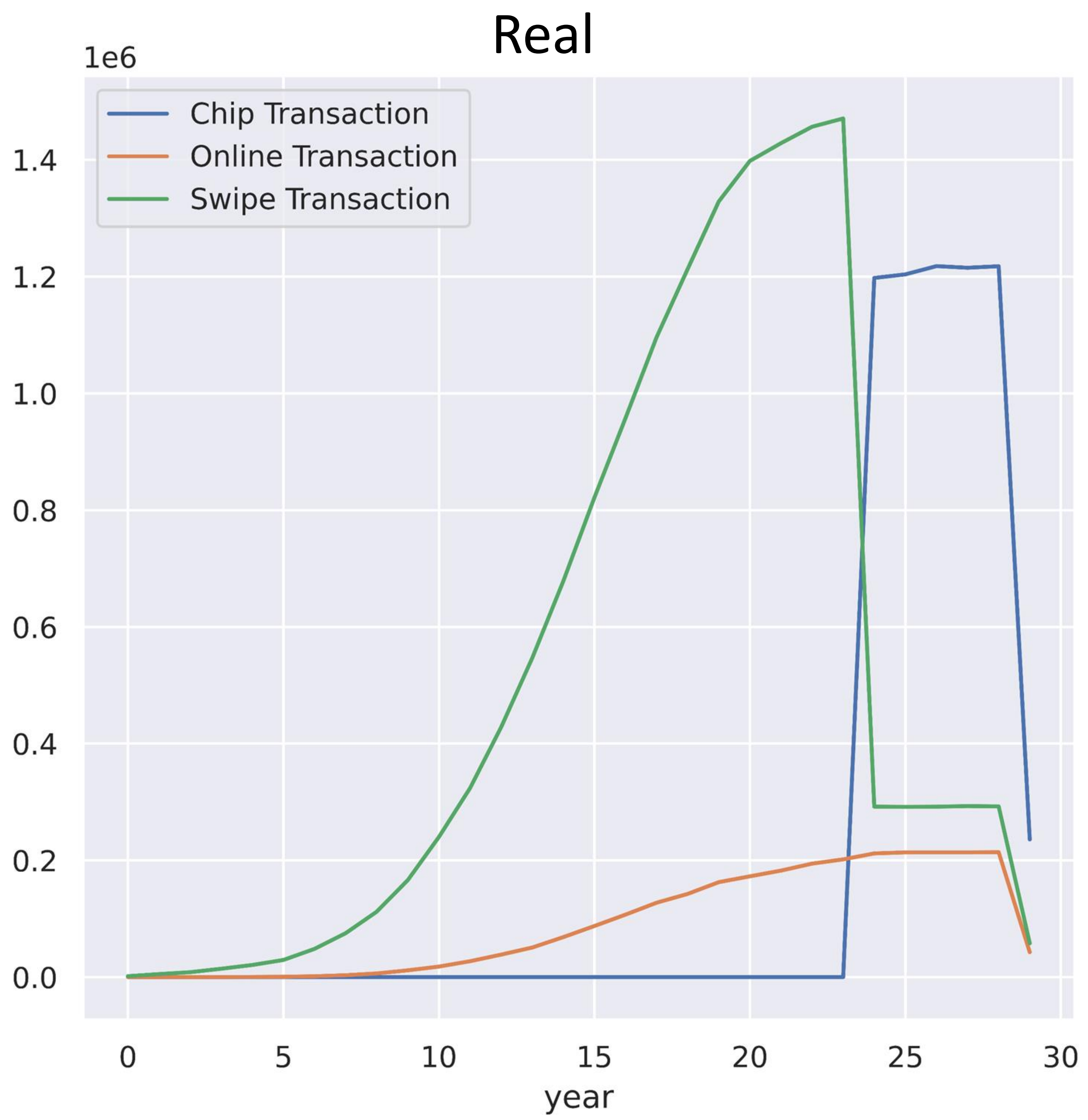
Real

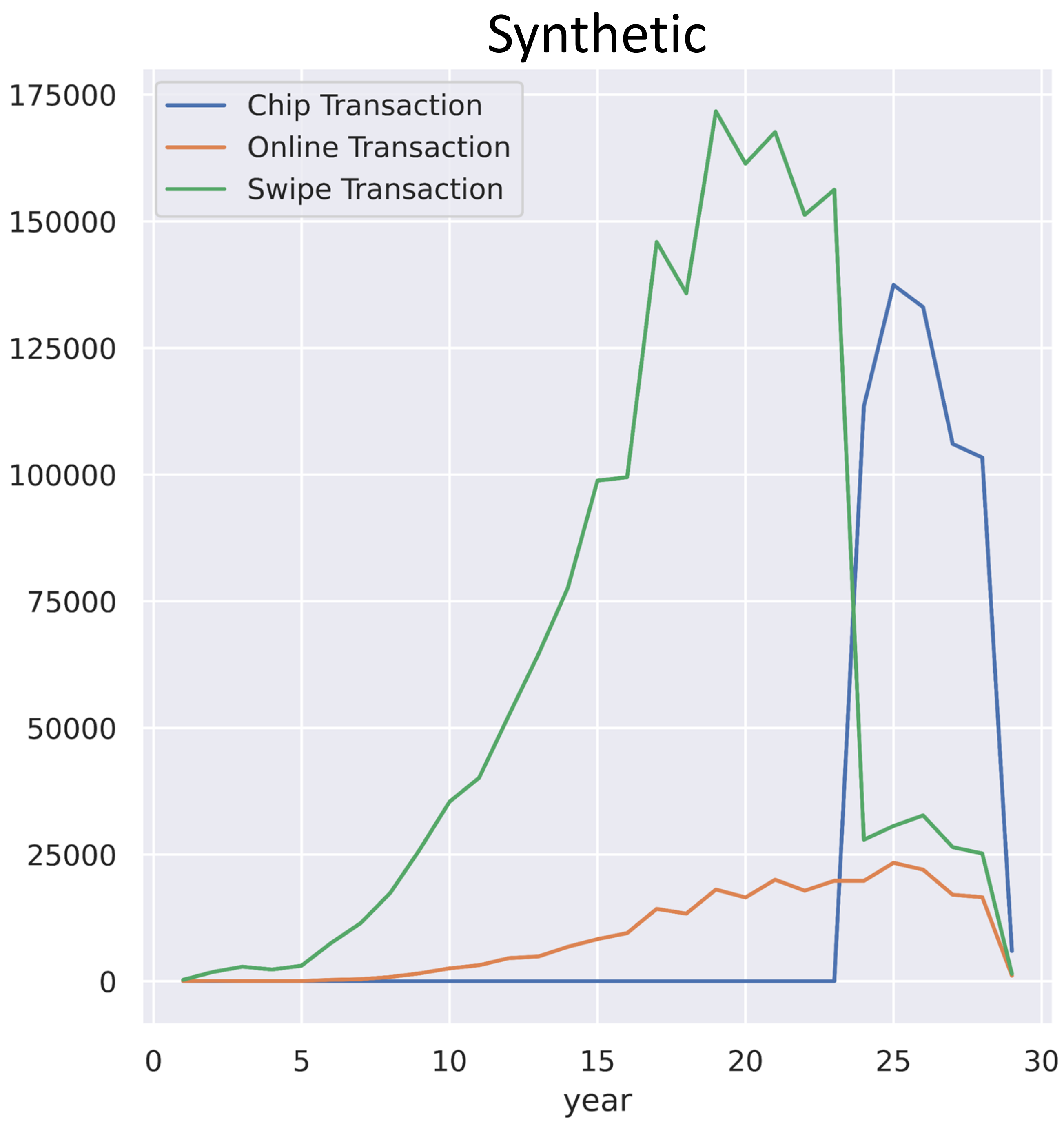


Synthetic

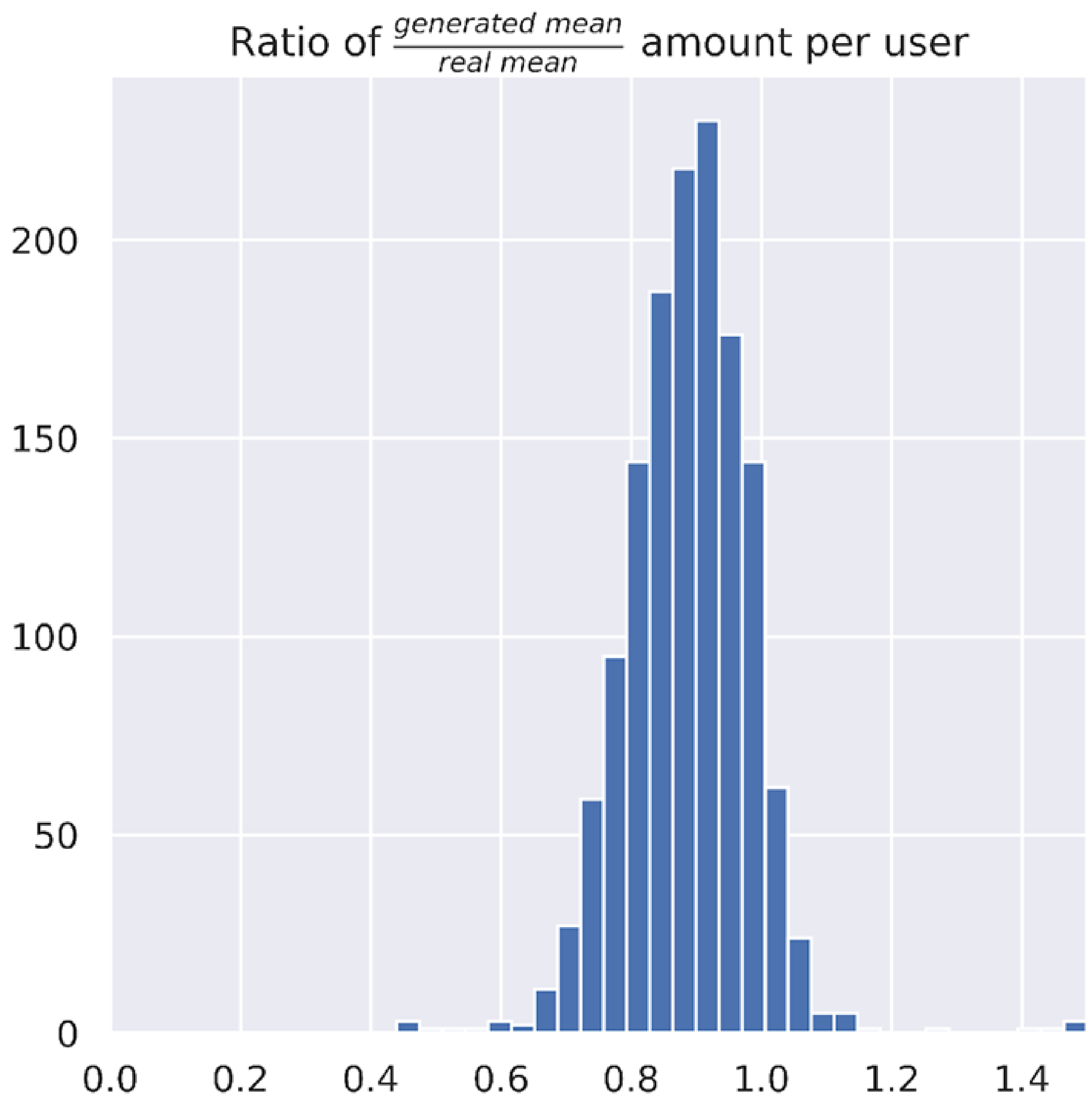


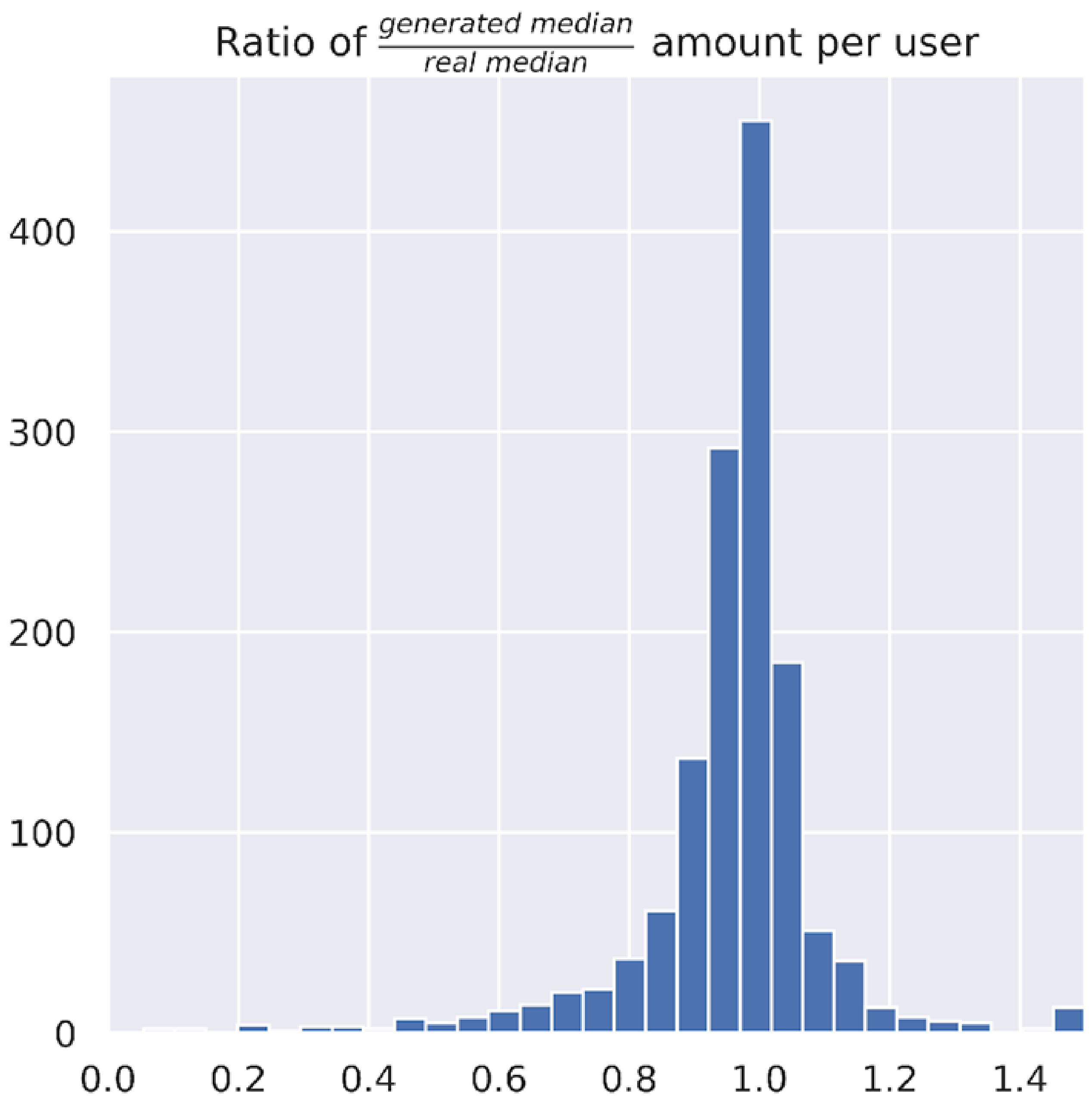
DOES IT WORK? USE CHIP BY YEAR





OVER ALL YEARS FOR A GIVEN USER, HOW DO THEIR REAL AND SYNTHETIC TRANSACTION AMOUNTS MATCH UP?





HOW FAR APART ARE TWO DISTRIBUTIONS?

Many names, one concept

Transport Problem

Earth Mover's Distance

Wasserstein Metric

1^{st} Mallows Distance

This can be formalized as the following linear programming problem: Let $P = \{(p_1, w_{p_1}), \ldots, (p_m, w_{p_m})\}$ be the first signature with *m* clusters, where p_i is the cluster representative and w_{p_i} is the weight of the cluster; $Q = \{(q_1, w_{q_1}), \ldots, (q_n, w_{q_n})\}$ the second signature with *n* clusters; and $\mathbf{D} = [d_{ij}]$ the ground distance matrix where d_{ij} is the ground distance between clusters p_i and q_j.

We want to find a flow $\mathbf{F} = [f_{ij}]$, with f_{ij} the flow between p_i and q_j, that minimizes the overall cost

$$\mathrm{WORK}(P, Q, \mathbf{F}) = \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij} \ ,$$

subject to the following constraints:

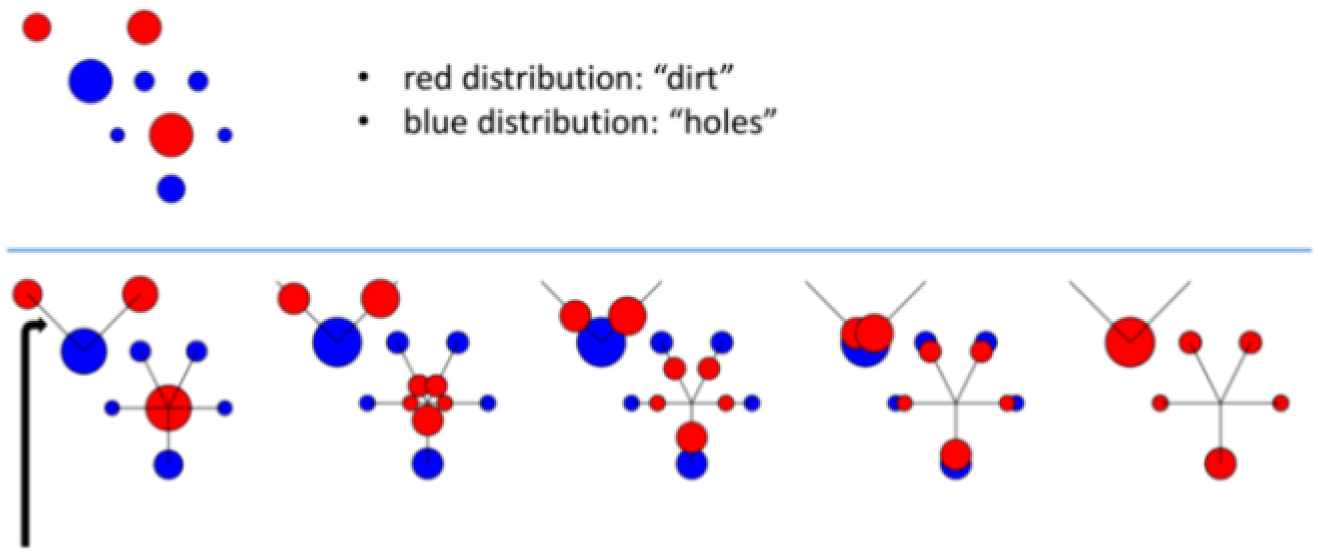
$$\begin{aligned} f_{ij} &\geq 0 \qquad 1 \leq i \leq m,\ 1 \leq j \leq n \\ \sum_{j=1}^{n} f_{ij} &\leq w_{p_i} \qquad 1 \leq i \leq m \\ \sum_{i=1}^{m} f_{ij} &\leq w_{q_j} \qquad 1 \leq j \leq n \\ \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} &= \min\Big(\sum_{i=1}^{m} w_{p_i}, \sum_{j=1}^{n} w_{q_j}\Big) \ , \end{aligned}$$

The first constraint allows moving ``supplies'' from *P* to *Q* and not vice versa. The next two constraints limits the amount of supplies that can be sent by the clusters in *P* to their weights, and the clusters in *Q* to receive no more supplies than their weights; and the last constraint forces to move the maximum amount of supplies possible. We call this amount the *total flow*. Once the transportation problem is solved, and we have found the optimal flow $\mathbf{F}$, the earth mover's distance is defined as the work normalized by the total flow:

$$\mathrm{EMD}(P, Q) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}} \ .$$

The normalization factor is introduced in order to avoid favoring smaller signatures in the case of partial matching.

https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/RUBNER/emd.htm



The distance between points (ground distance) can be Euclidean distance, Manhattan...

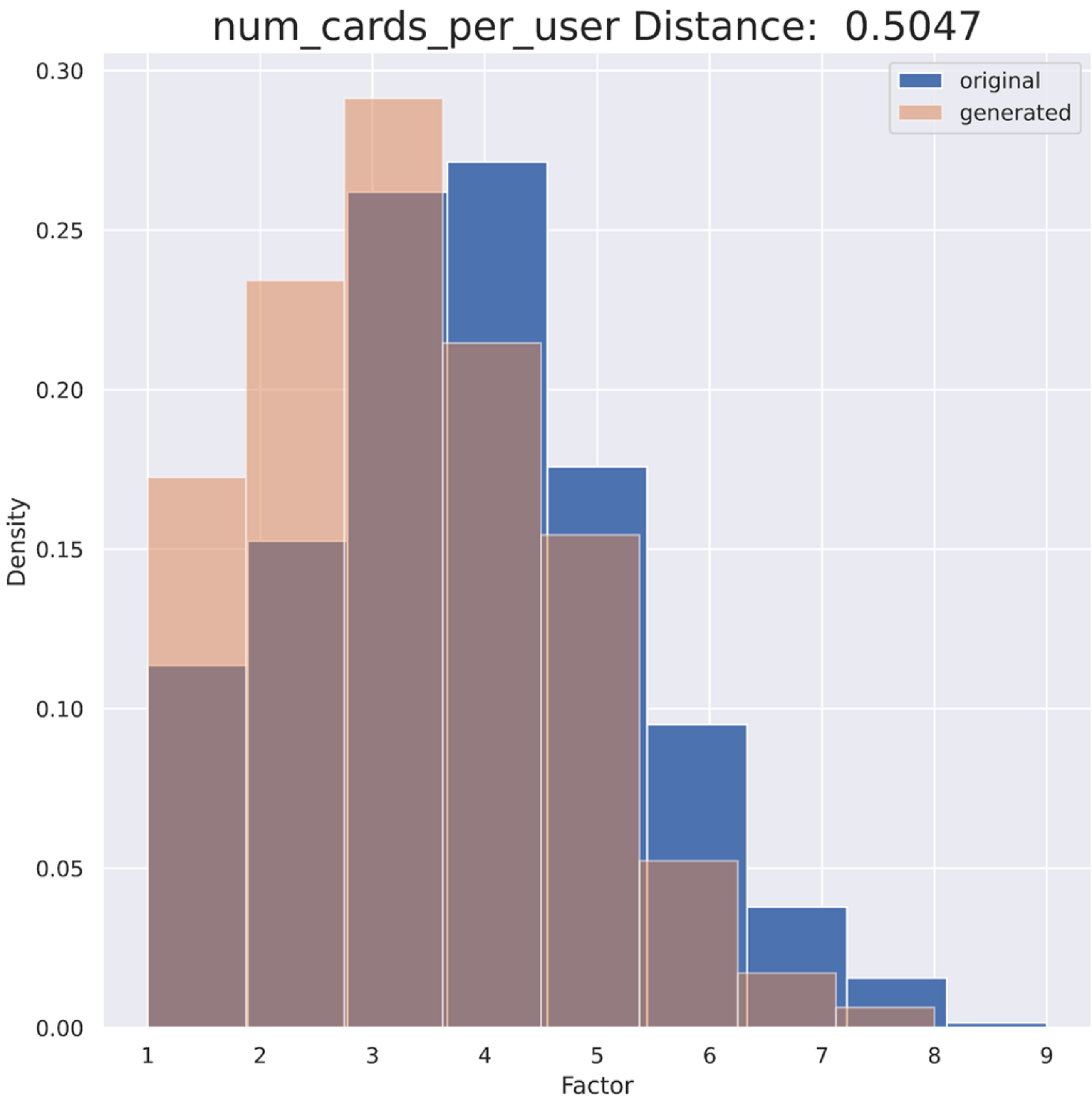
Example 1

The goal of the EMD algorithm is to optimize how to distribute the weights so that all of the dirt covers all of the holes while moving the weights through the minimum distance possible.

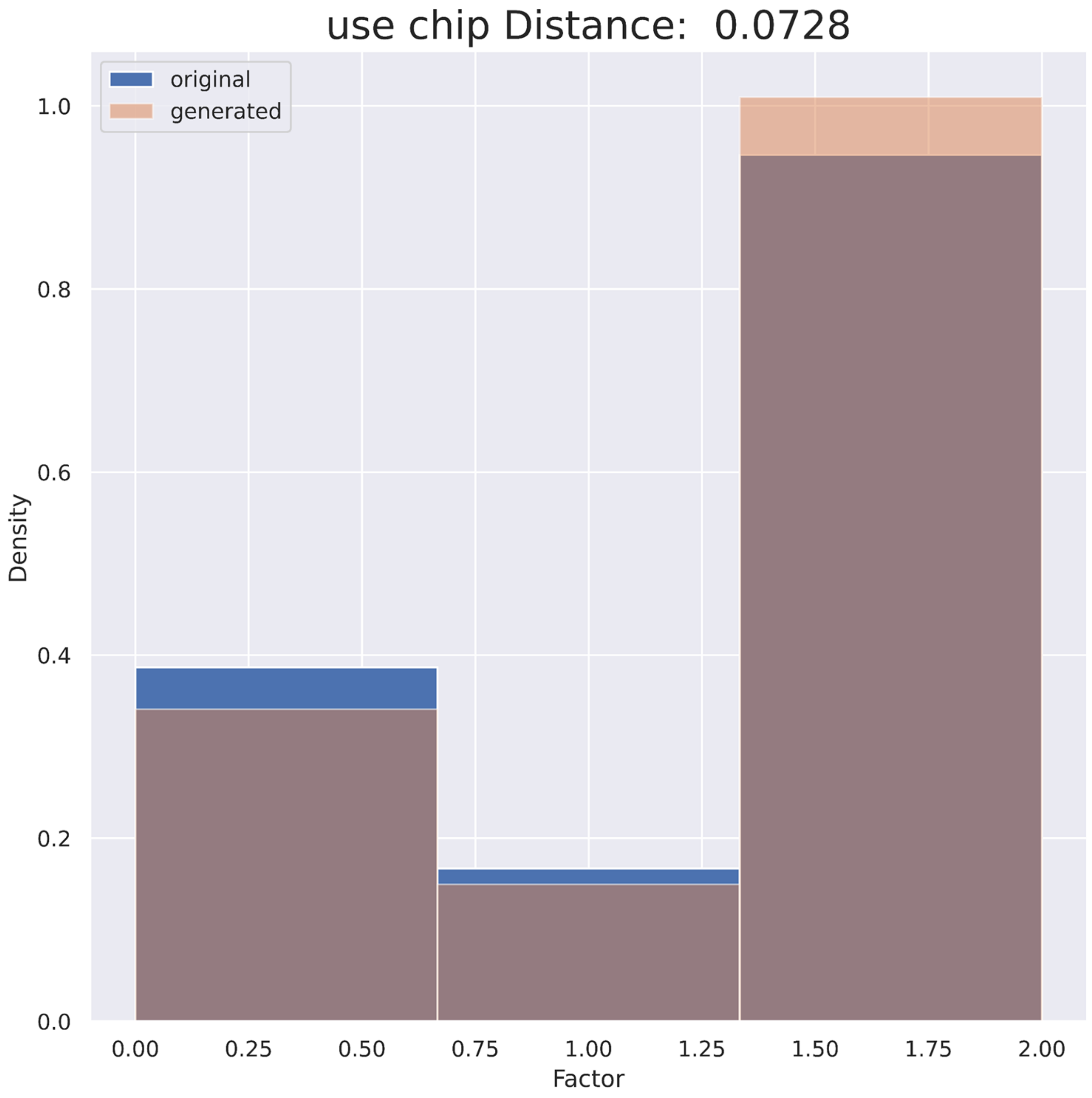
https://towardsdatascience.com/earth-movers-distance-68fff0363ef2

NVIDIA.

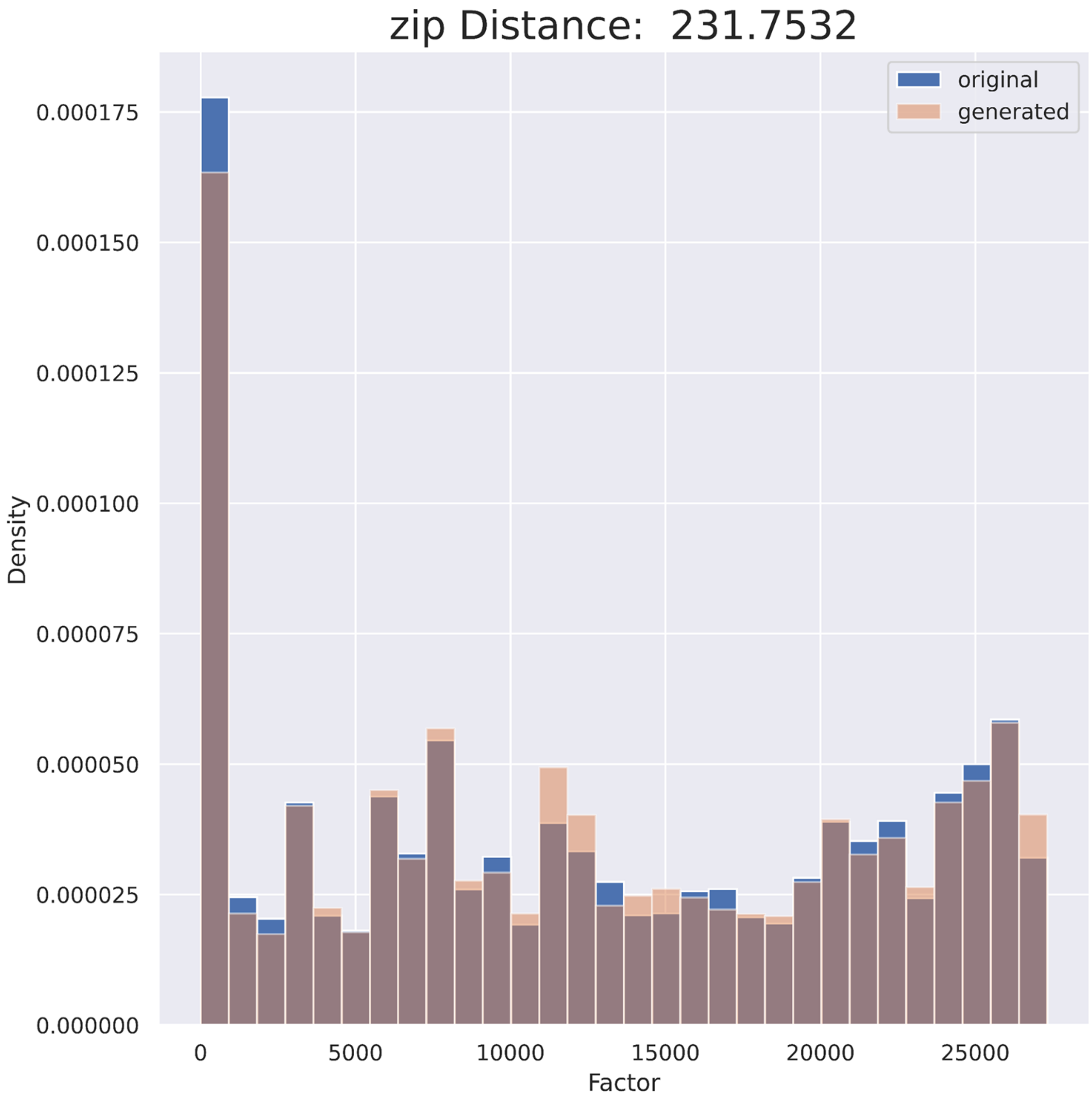
HISTOGRAMS & DISTANCES



HISTOGRAMS

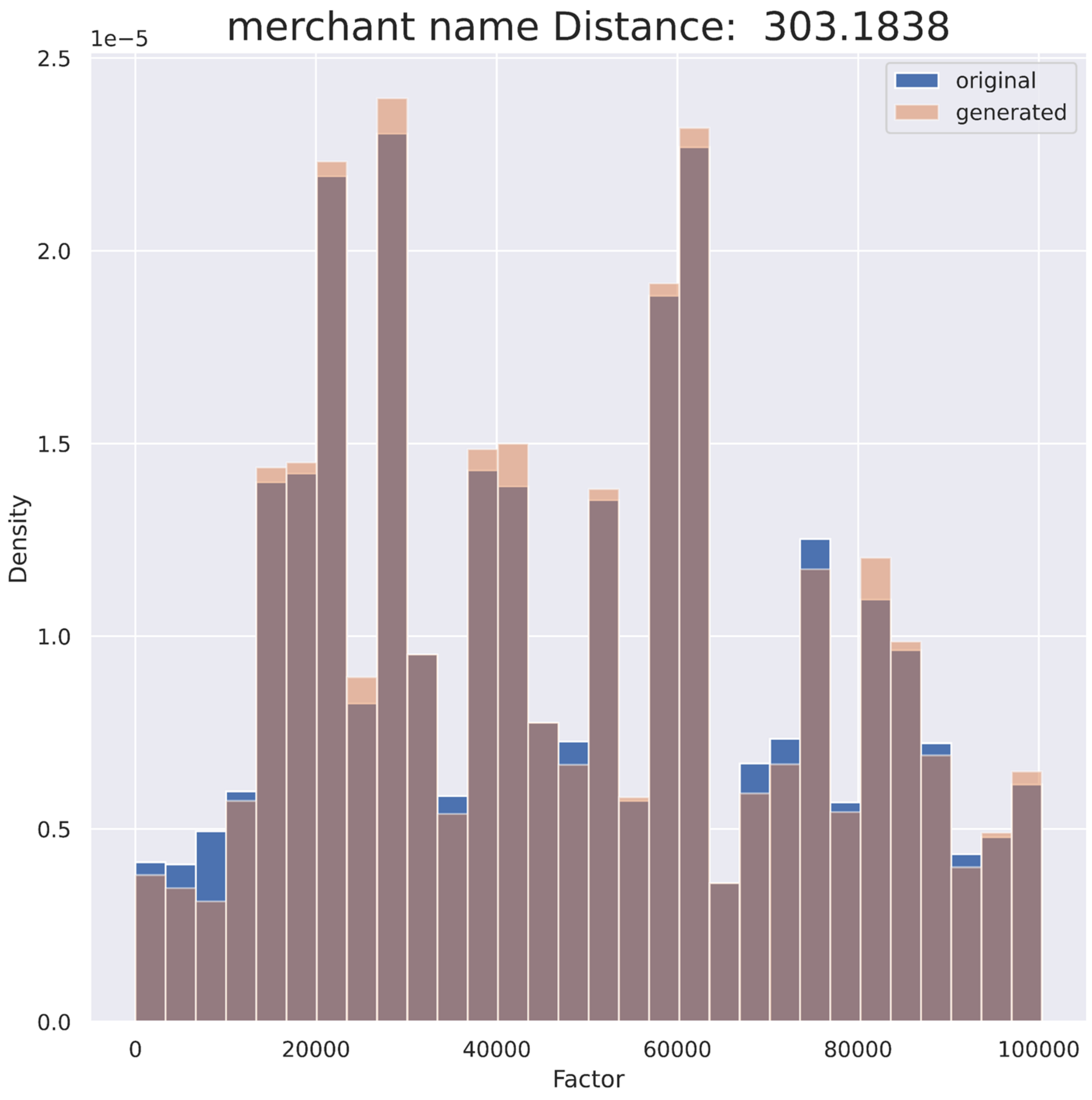


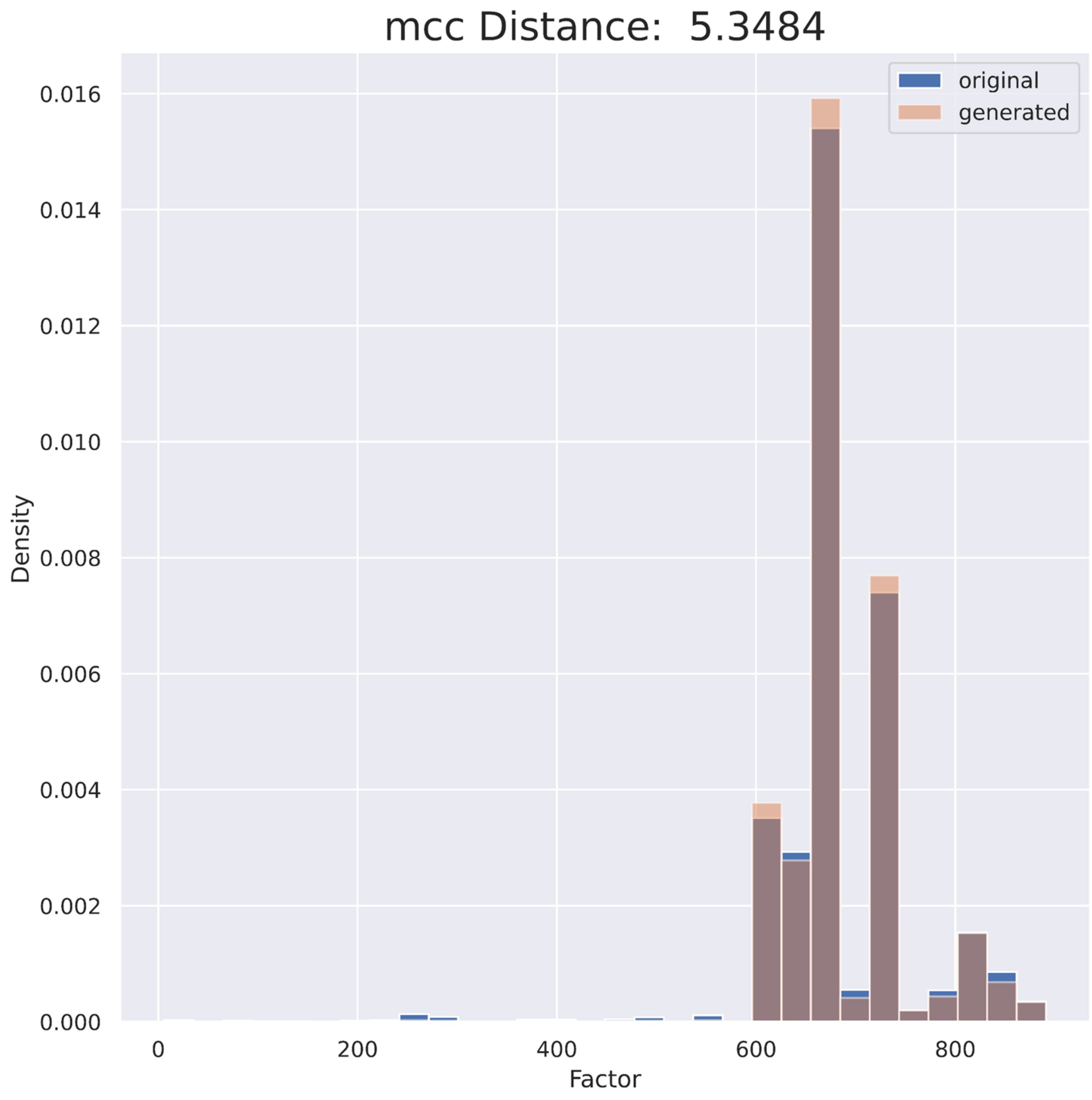
use chip Distance: 0.0728
original
generated
1.0
0.8
0.6
0.4
0.2
0.0
Density
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Factor



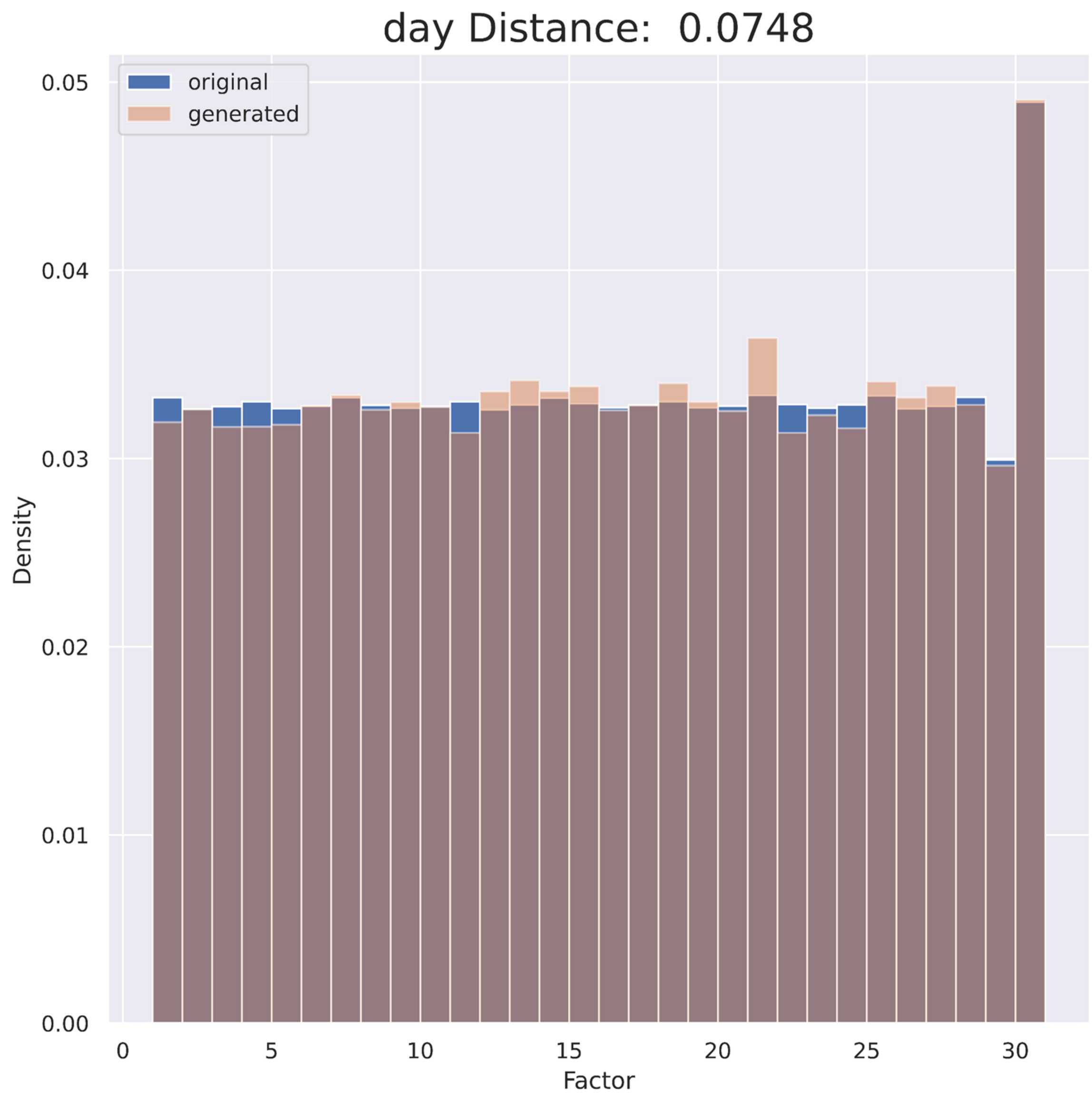
zip Distance: 231.7532
original
generated
0.000175
0.000150
0.000125
0.000100
0.000075
0.000050
0.000025
0.000000
Density
0
5000
10000
15000
20000
25000
Factor

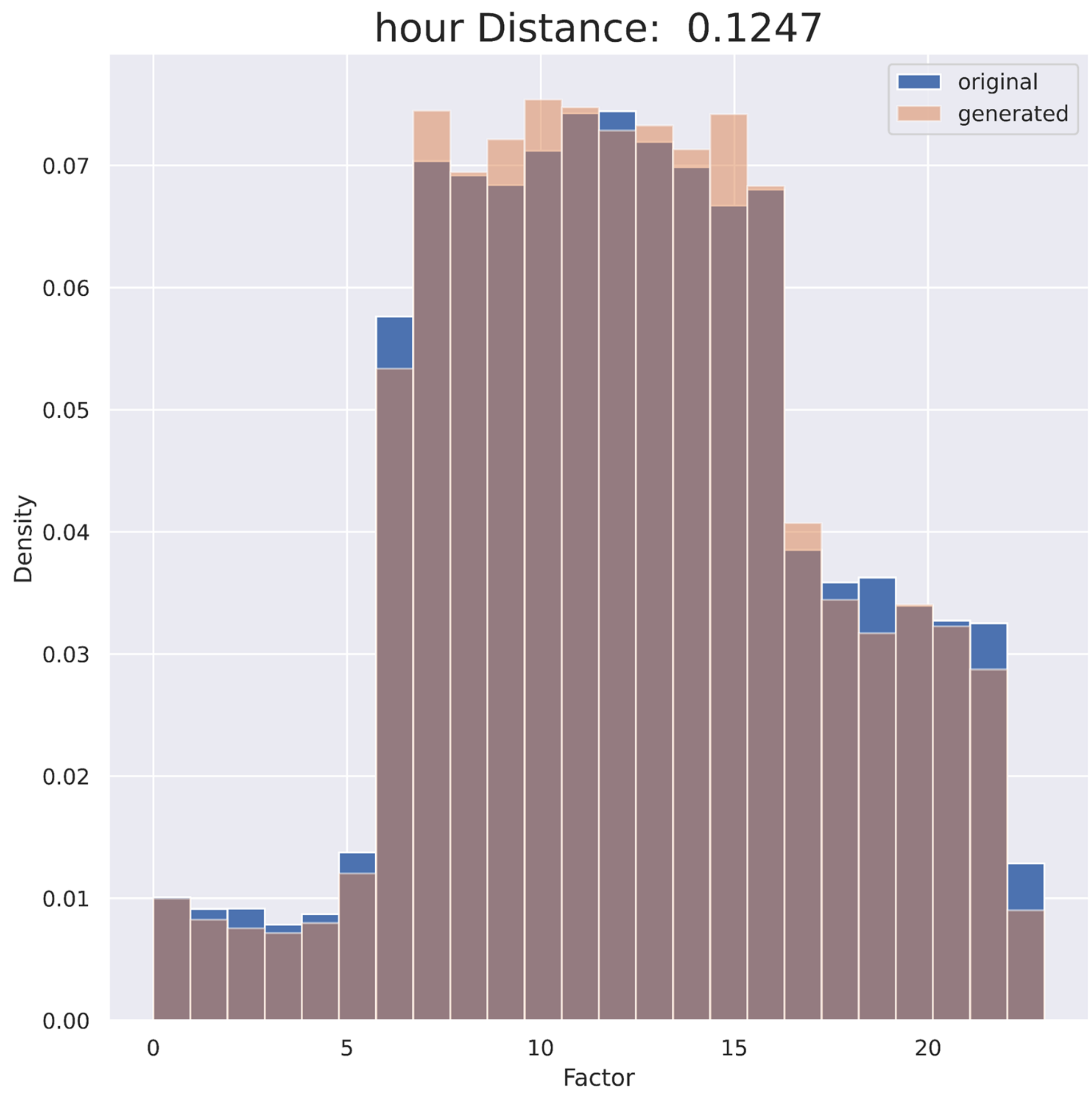
HISTOGRAMS

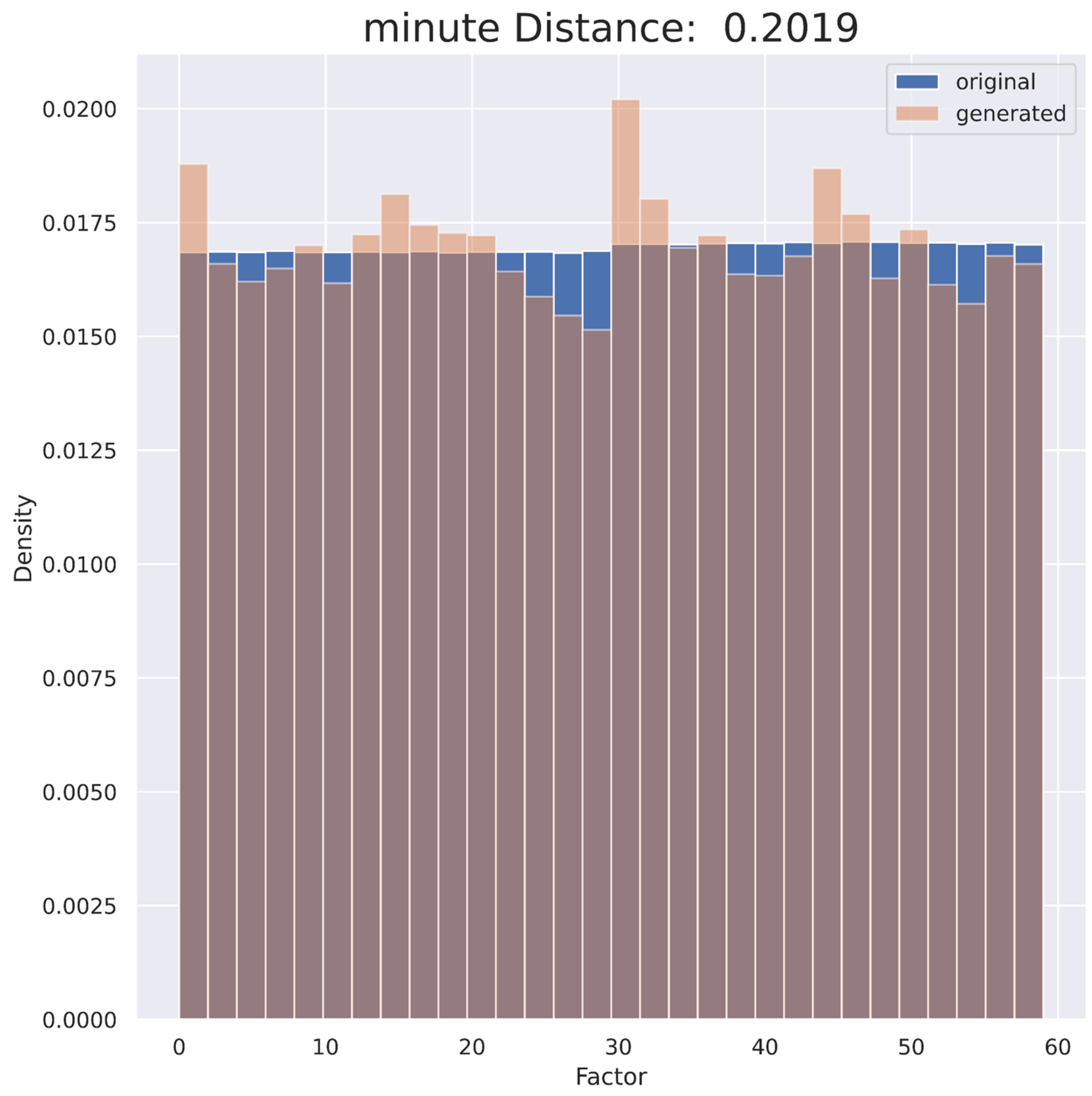

merchant name Distance: 303.1838
1e−5
2.5
2.0
1.5
1.0
0.5
0.0
Density
0
20000
40000
60000
80000
100000
Factor
original
generated


mcc Distance: 5.3484
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
Density
0
200
400
600
800
Factor
original
generated

HISTOGRAMS

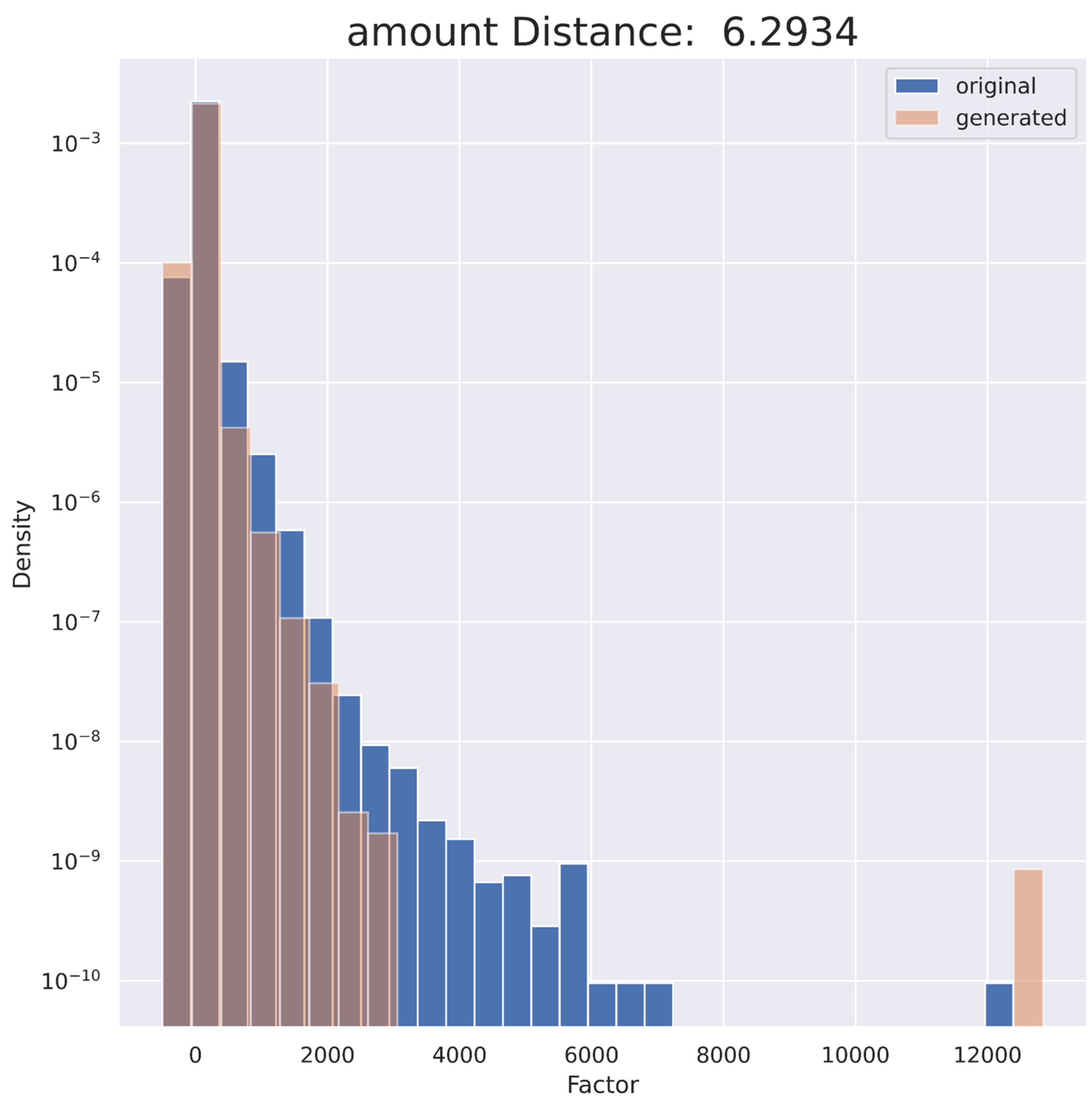

day Distance: 0.0748
original
generated
Density
Factor
0.00
0.01
0.02
0.03
0.04
0.05
0
5
10
15
20
25
30

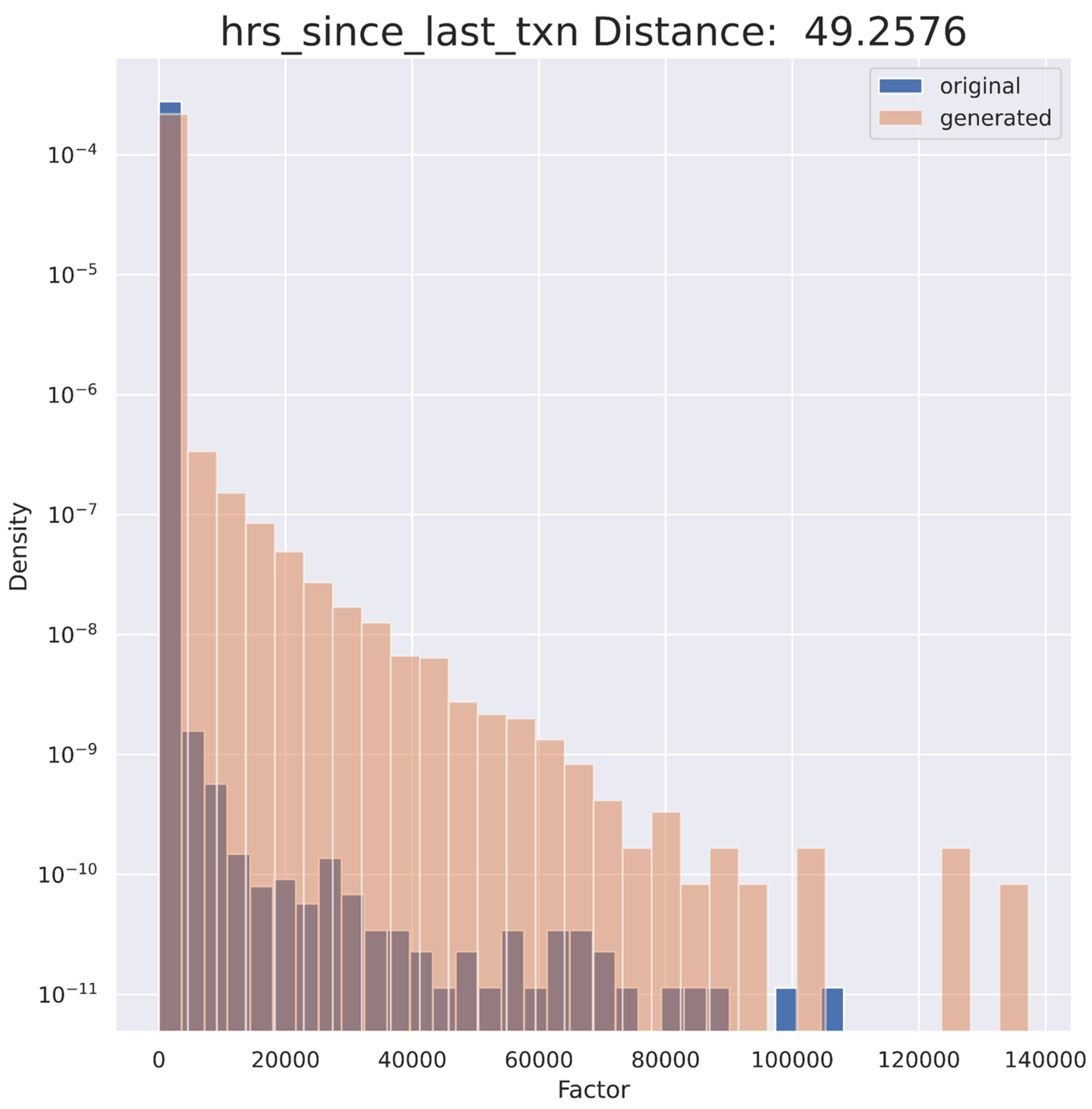

hour Distance: 0.1247
original
generated
Density
Factor
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0
5
10
15
20


minute Distance: 0.2019
original
generated
Density
Factor
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
0.0200
0
10
20
30
40
50
60

MORE WORK TO BE DONE!

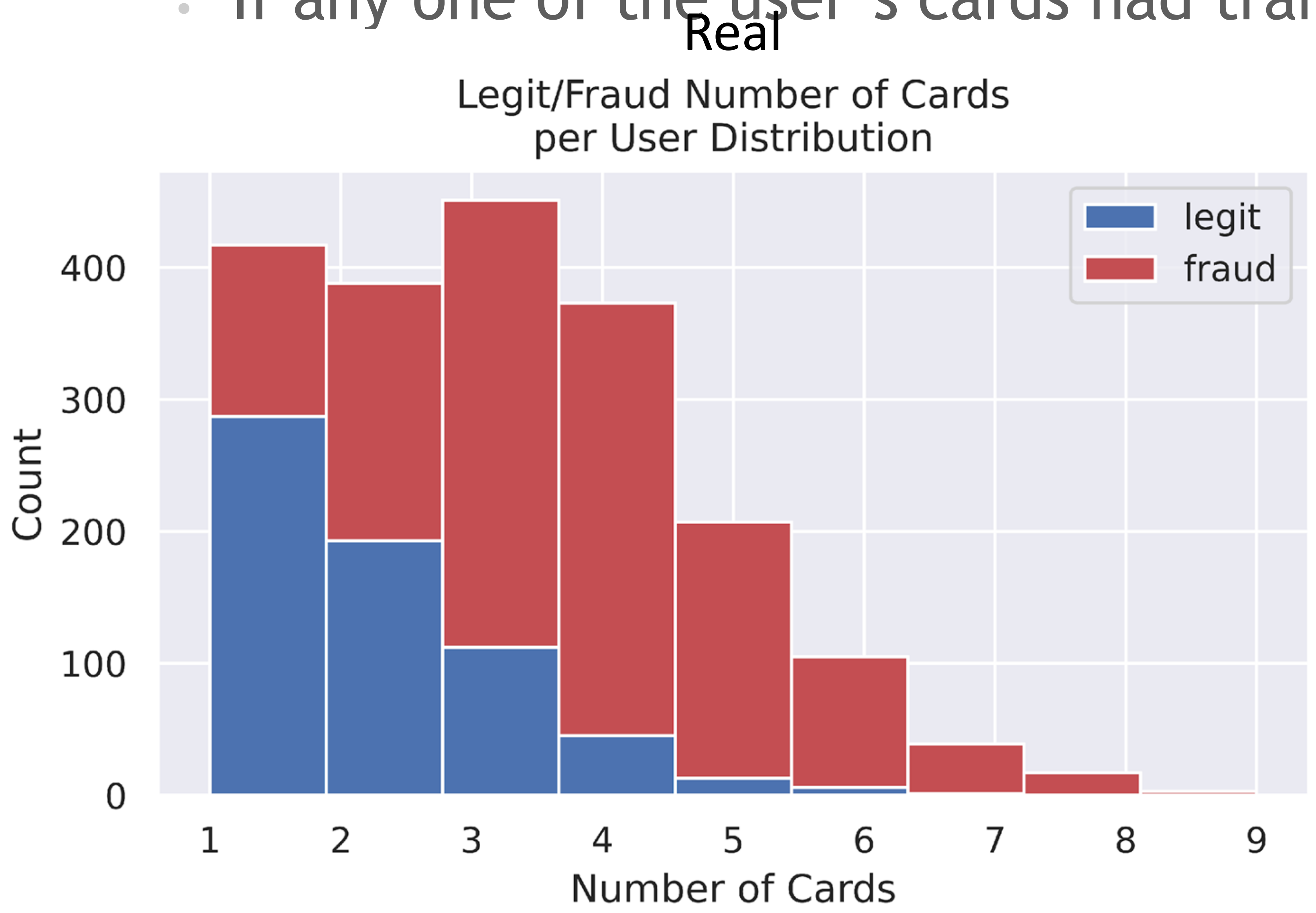
HISTOGRAMS (LOG SCALE)

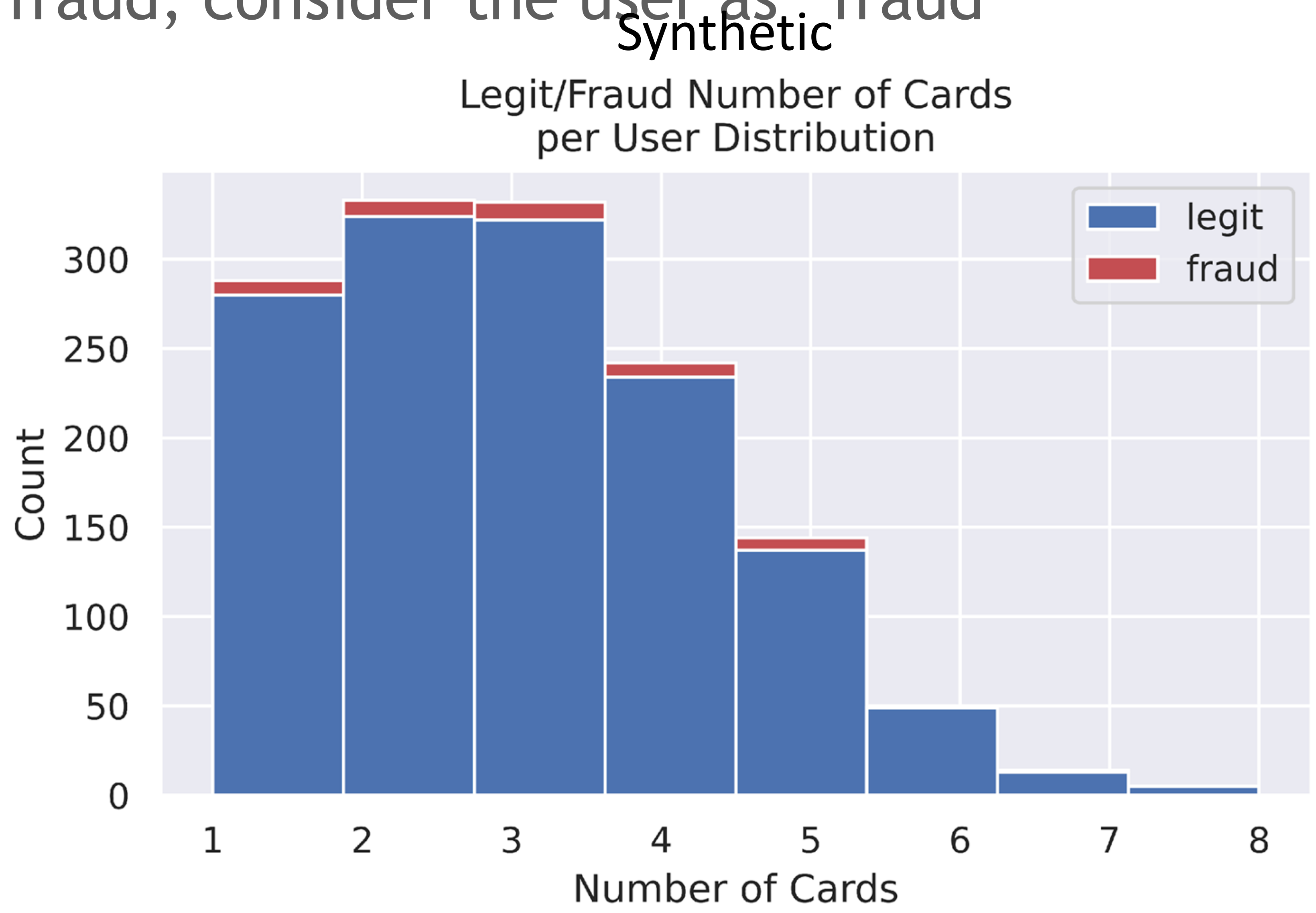




FRAUD DISTRIBUTION PER USER-CARD ALTHOUGH THIS MAY MAKE THE DATASET MORE REALISTIC...

- Question: “What is distribution of legit/fraud for users with X number of cards?”
  - Assume all fraud labels in the original dataset are true positives.
  - If any one of the user’s cards had transaction fraud, consider the user as “fraud”





AMOUNT VARIETY

- 99.97% transaction amounts were found in the original data
- Of the remaining 0.03%, 92.5% of the amounts were within 5 cents of an original amount in the dataset
- The max difference of a generated amount from any observed amount was $451.61

RETURNS OCCUR BEFORE A PURCHASE

- Example
- Make return for $65 then purchase $65 ☒
- Same merchant name! ☑

user	card	year	month	day	hour	minute	amount	use chip	merchant name	zip	mcc	is_fraud	date
1490	0	1	8	7	14	6	-65.0	2	59935	26964	677	0	1992-08-07 14:06:00
1490	0	1	8	7	14	17	65.0	2	59935	26964	677	0	1992-08-07 14:17:00

FURTHER READING

arXiv:2005.14165v4 [cs.CL] 22 Jul 2020

Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*

Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry

Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan

Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter

Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray

Benjamin Chess Jack Clark Christopher Berner

Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI

Abstract

Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, humans can generally perform a new language task from only a few examples or from simple instructions – something which current NLP systems still largely struggle to do. Here we show that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching competitiveness with prior state-of-the-art fine-tuning approaches. Specifically, we train GPT-3, an autoregressive language model with 175 billion parameters, 10x more than any previous non-sparse language model, and test its performance in the few-shot setting. For all tasks, GPT-3 is applied without any gradient updates or fine-tuning, with tasks and few-shot demonstrations specified purely via text interaction with the model. GPT-3 achieves strong performance on many NLP datasets, including translation, question-answering, and cloze tasks, as well as several tasks that require on-the-fly reasoning or domain adaptation, such as unscrambling words, using a novel word in a sentence, or performing 3-digit arithmetic. At the same time, we also identify some datasets where GPT-3's few-shot learning still struggles, as well as some datasets where GPT-3 faces methodological issues related to training on large web corpora. Finally, we find that GPT-3 can generate samples of news articles which human evaluators have difficulty distinguishing from articles written by humans. We discuss broader societal impacts of this finding and of GPT-3 in general.

arXiv:2102.07350v1 [cs.CL] 15 Feb 2021

Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm

Laria Reynolds
moire@knc.ai

Kyle McDonell
kyle@knc.ai

Abstract

Prevailing methods for mapping large generative language models to supervised tasks may fail to sufficiently probe models' novel capabilities. Using GPT-3 as a case study, we show that 0-shot prompts can significantly outperform few-shot prompts. We suggest that the function of few-shot examples in these cases is better described as locating an already learned task rather than meta-learning. This analysis motivates rethinking the role of prompts in controlling and evaluating powerful language models. In this work, we discuss methods of prompt programming, emphasizing the usefulness of considering prompts through the lens of natural language. We explore techniques for exploiting the capacity of narratives and cultural anchors to encode nuanced intentions and techniques for encouraging deconstruction of a problem into components before producing a verdict. Informed by this more encompassing theory of prompt programming, we also introduce the idea of a *metaprompt* that seeds the model to generate its own natural language prompts for a range of tasks. Finally, we discuss how these more general methods of interacting with language models can be incorporated into existing and future benchmarks and practical applications.

Keywords: language models, transformers, GPT-3, few-shot learning, prompt programming, metaprompts, serial reasoning, semiotics

1 Motivation

The recent rise of massive self-supervised language models such as GPT-3 [3] and their success on downstream tasks has brought us one step closer to the goal of task-agnostic artificial intelligence systems. However, despite the apparent power of such models, current methods of controlling them to perform specific tasks are extremely limited. In order to properly evaluate their capabilities and extract useful work from these models, new methods are required.

mat at extracting specific learned behaviors from self-supervised language models.

We argue that contrary to the common interpretation of the few-shot format implied by the title of the original GPT-3 paper [3], *Language models are few-shot learners*, GPT-3 is often not actually *learning* the task during run time from few-shot examples. Rather than instruction, the method's primary function is *task location* in the model's existing space of learned tasks. This is evidenced by the effectiveness of alternative prompts which, with no examples or instruction, can elicit comparable or superior performance to the few-shot format.

TRANSFORMER FOR NLP, VISION, AUDIO

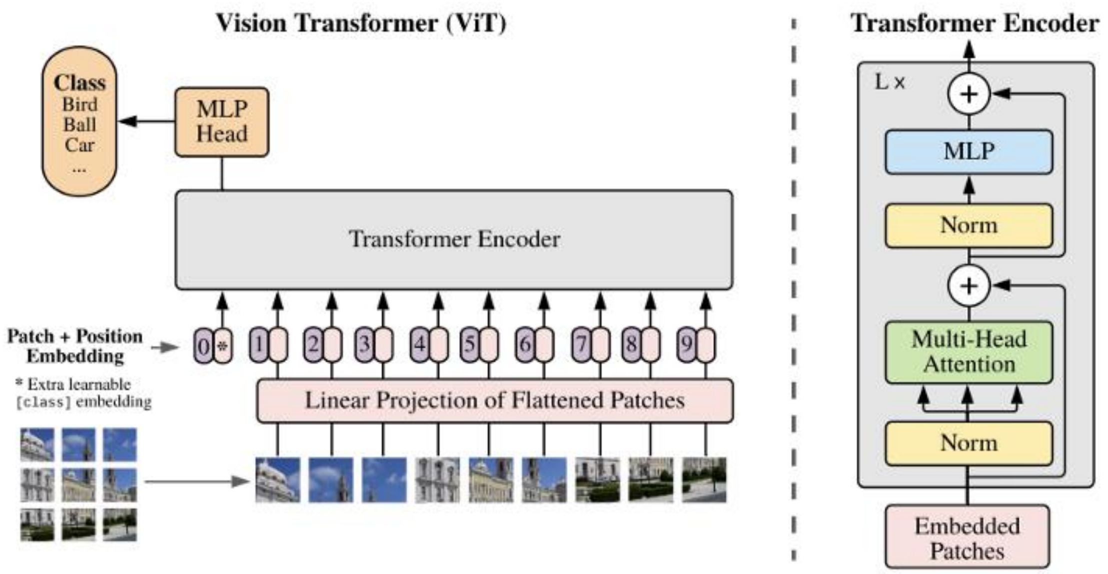
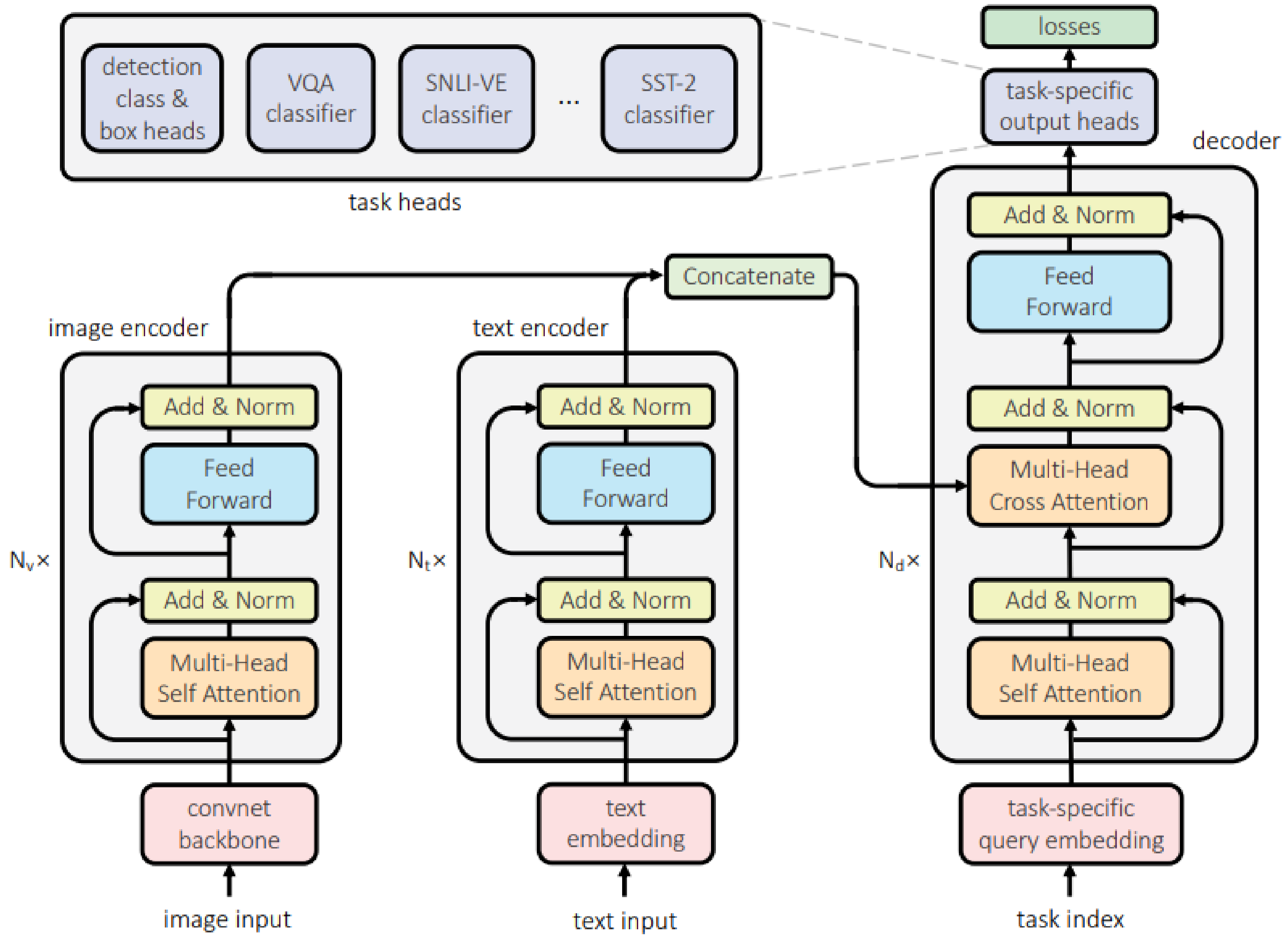
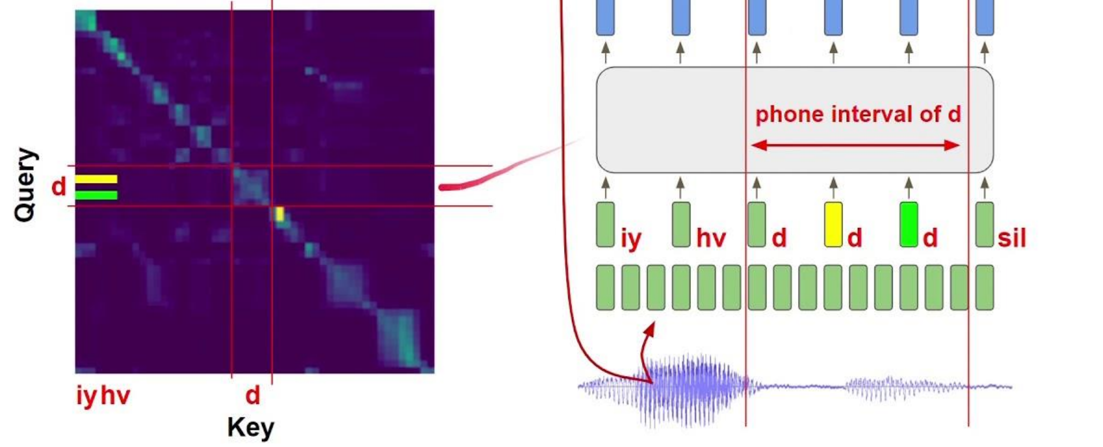


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).



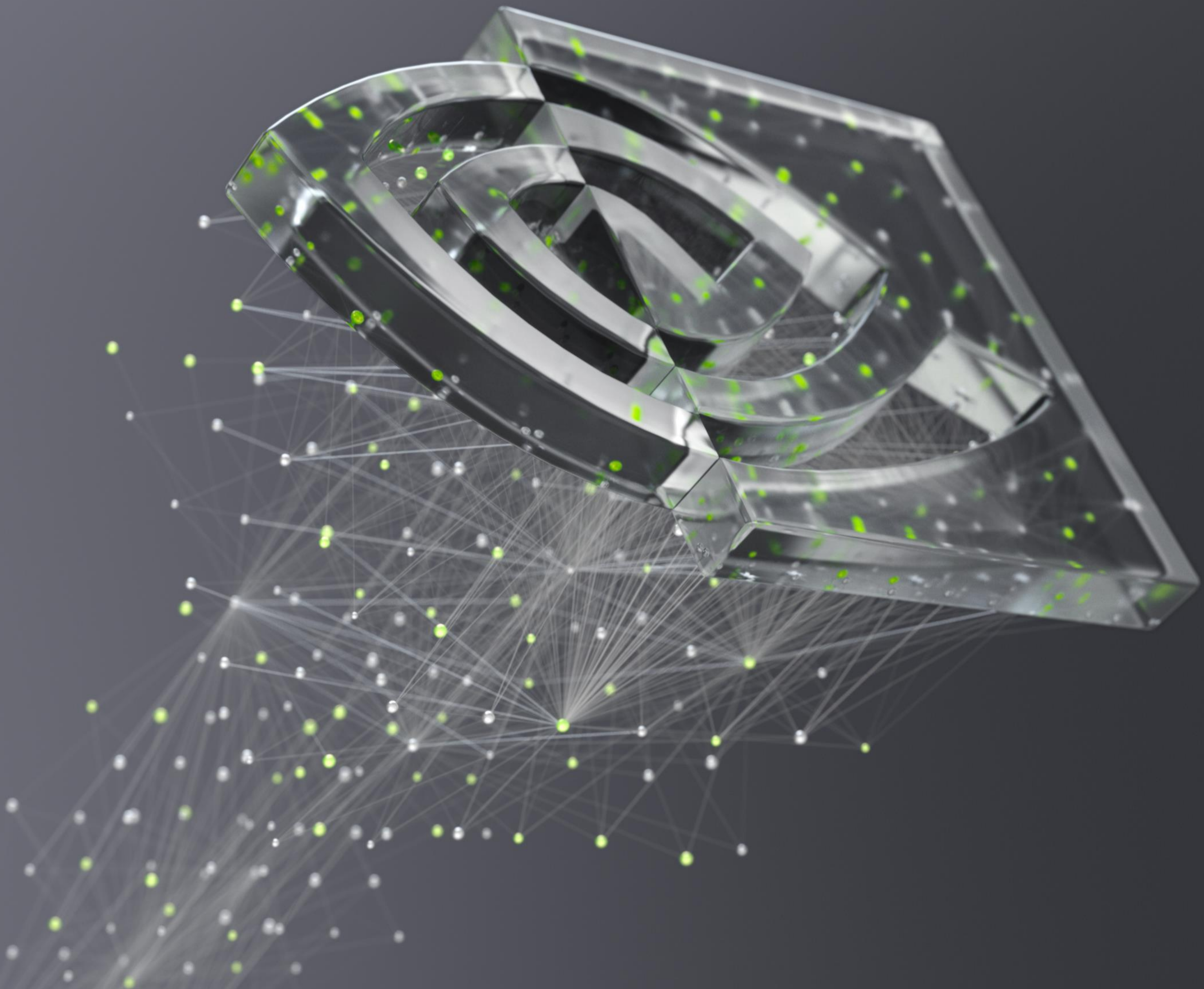
Diagonal attention

- Block diagonal



Transformer is All You Need:
Multimodal Multitask Learning with a Unified Transformer

Ronghang Hu   Amanpreet Singh
Facebook AI Research (FAIR)



NVIDIA