AI Infrastructure

Adam Grzywaczewski, Senior Deep Learning Data Scientist

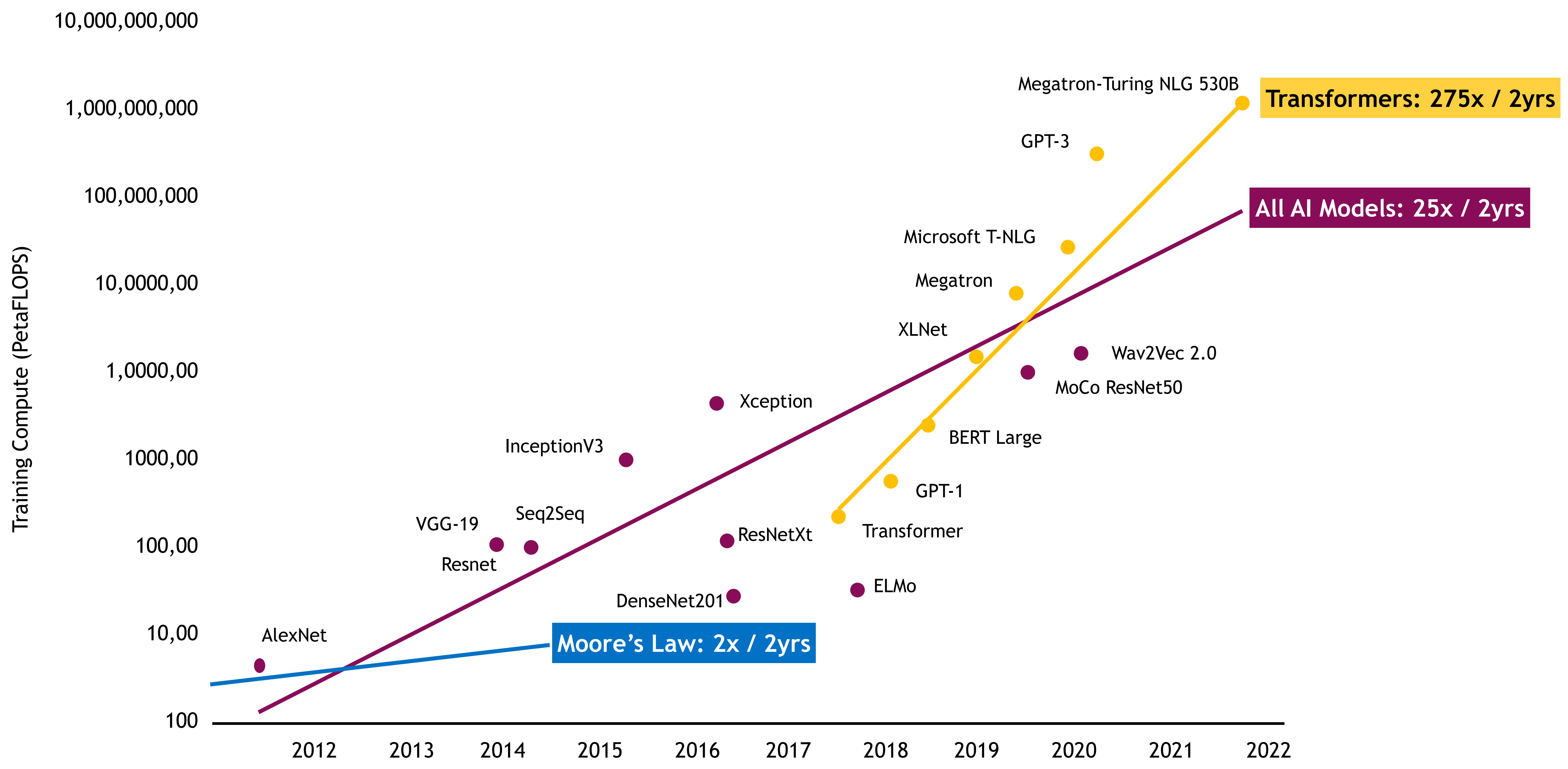
ABOUT ME

Adam Grzywaczewski – adamg@nvidia.com

- Senior Deep Learning Data Scientist @ NVIDIA - Supporting delivery of AI / Deep Learning solutions
- Focusing on large scale/distributed training and efficient inference
- Expertise in Natural Language Processing

DRAMATIC INCREASE IN MODEL SIZES

The Trend Continues



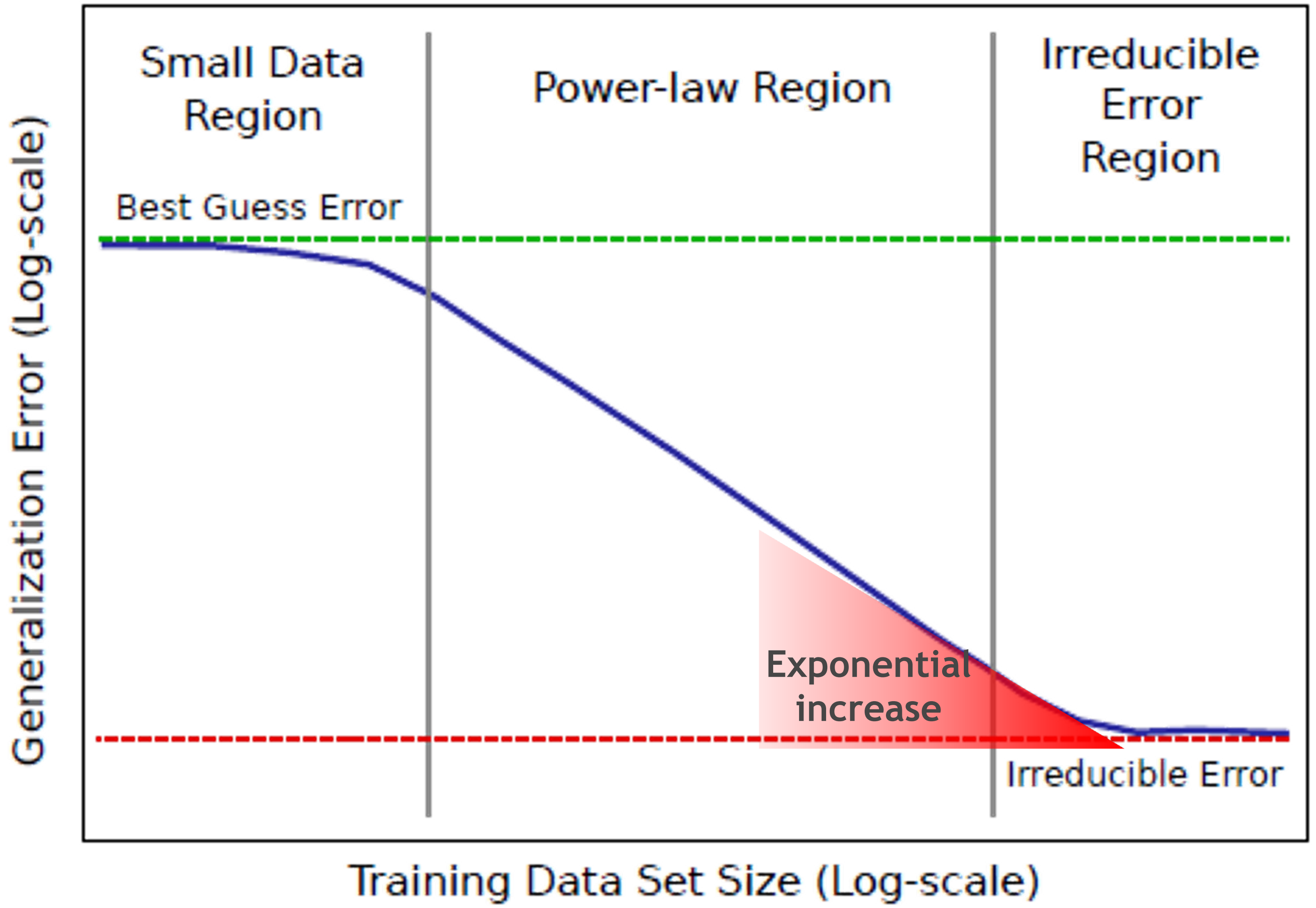
https://developer.nvidia.com/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model

NVIDIA

Why?

THE SCALING LAWS

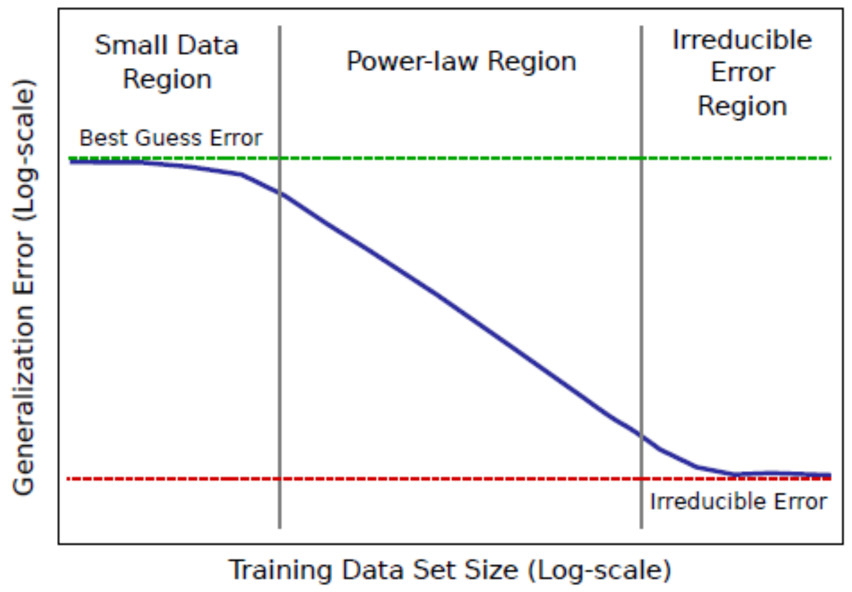
Performance of neural networks increases with model/dataset size

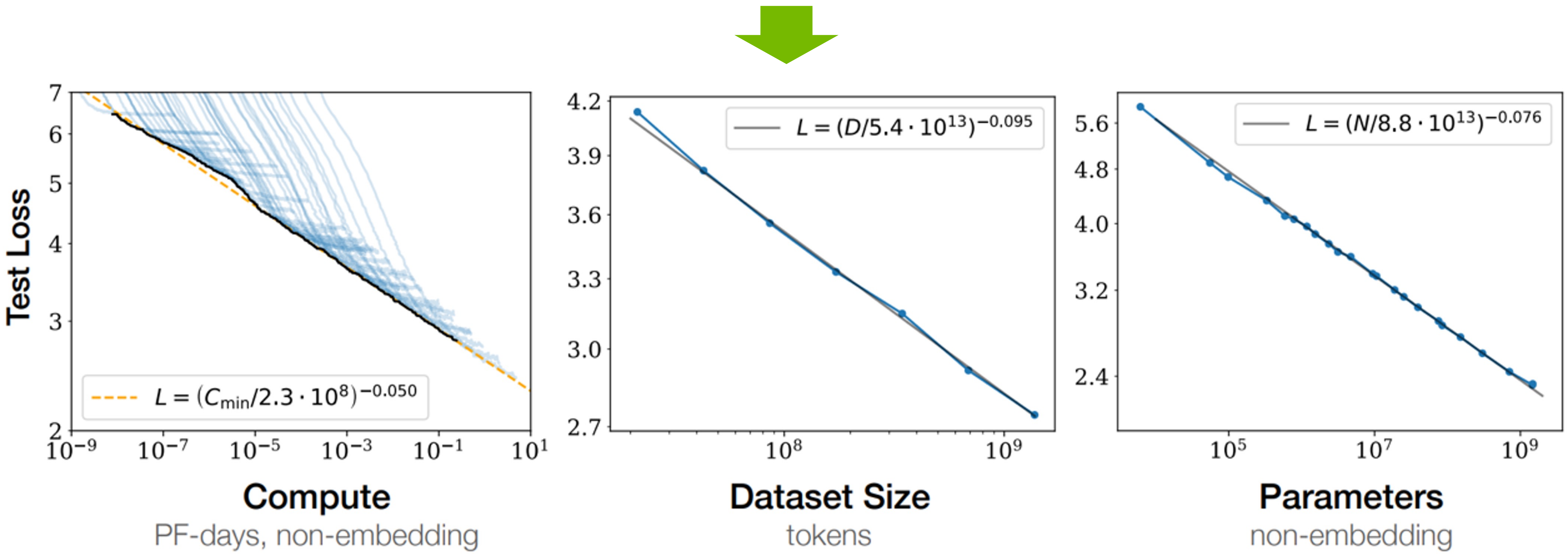


Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., ... & Zhou, Y. (2017). **Deep Learning Scaling is Predictable, Empirically**. arXiv:1712.00409

EMPIRICAL EVIDENCE

The Scaling Laws in NLP

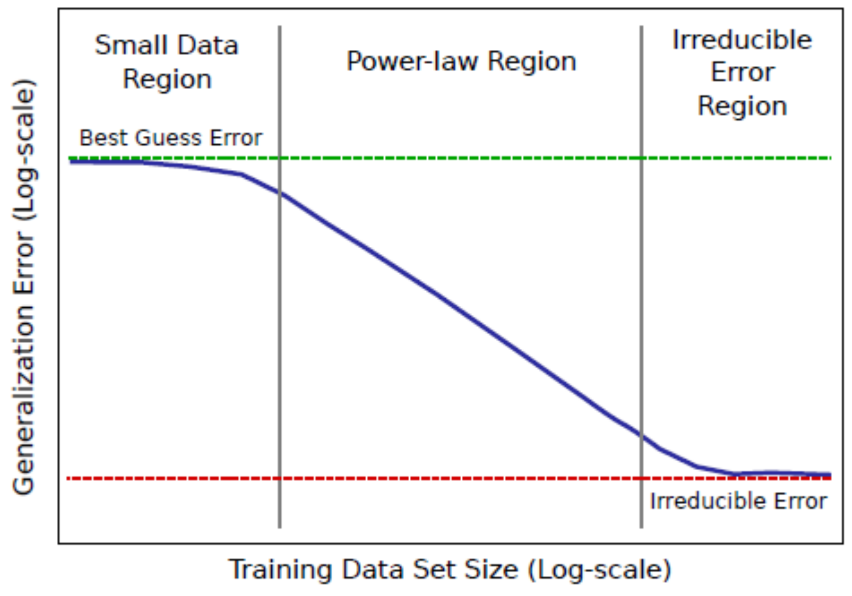


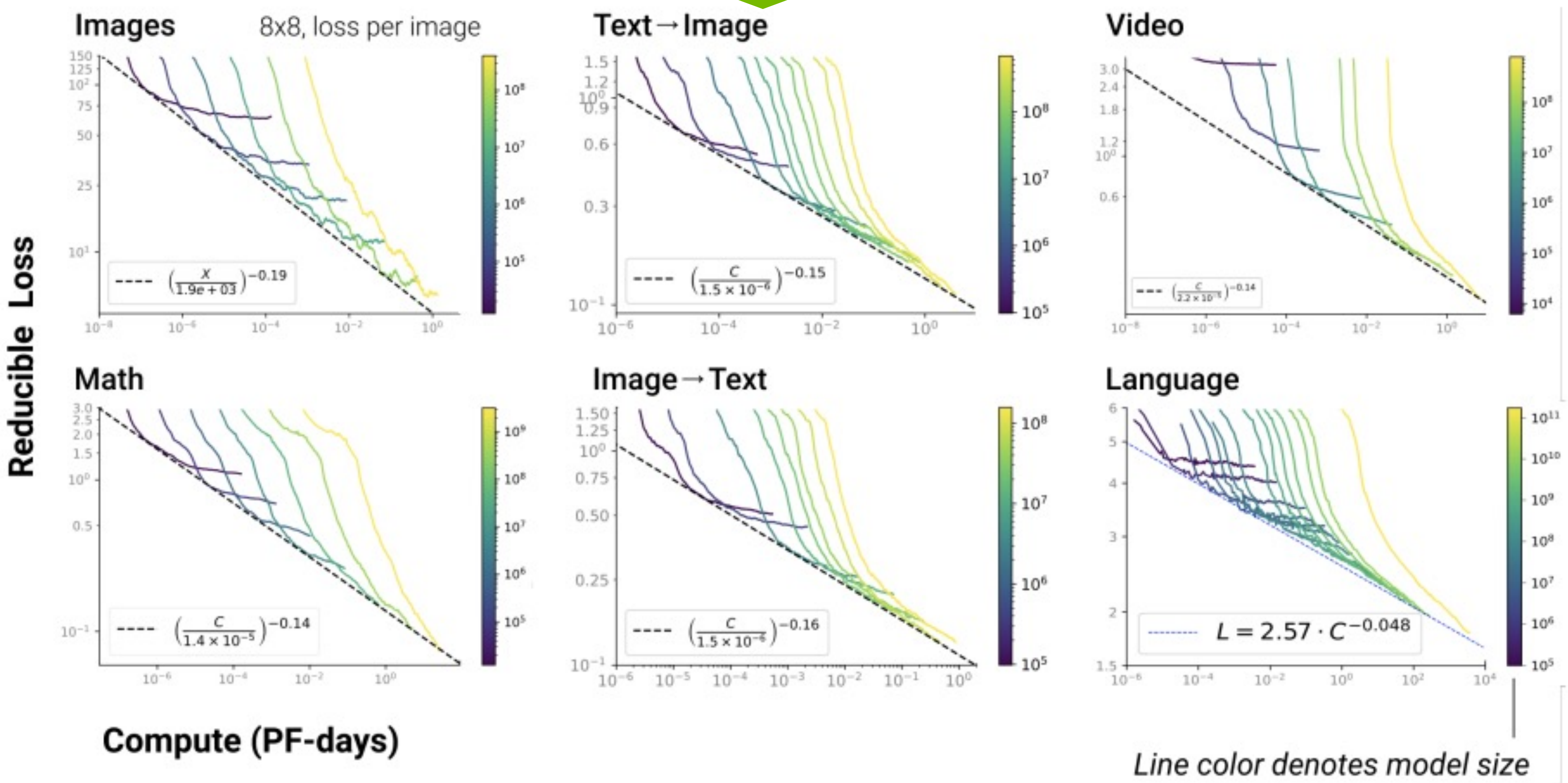


Henighan, Tom, et al. **Scaling laws for autoregressive generative modeling.** arXiv preprint arXiv:2010.14701 (2020).

EMPIRICAL EVIDENCE

The Scaling Laws for Generative models

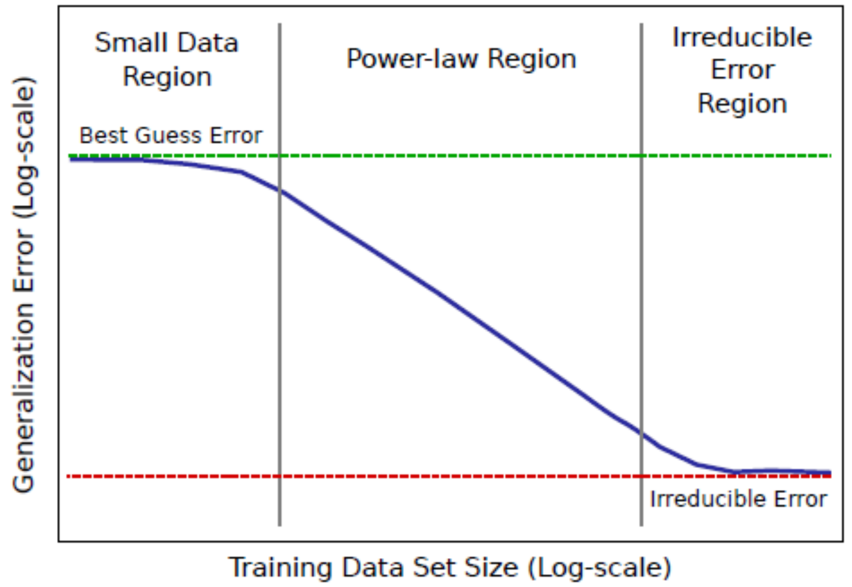


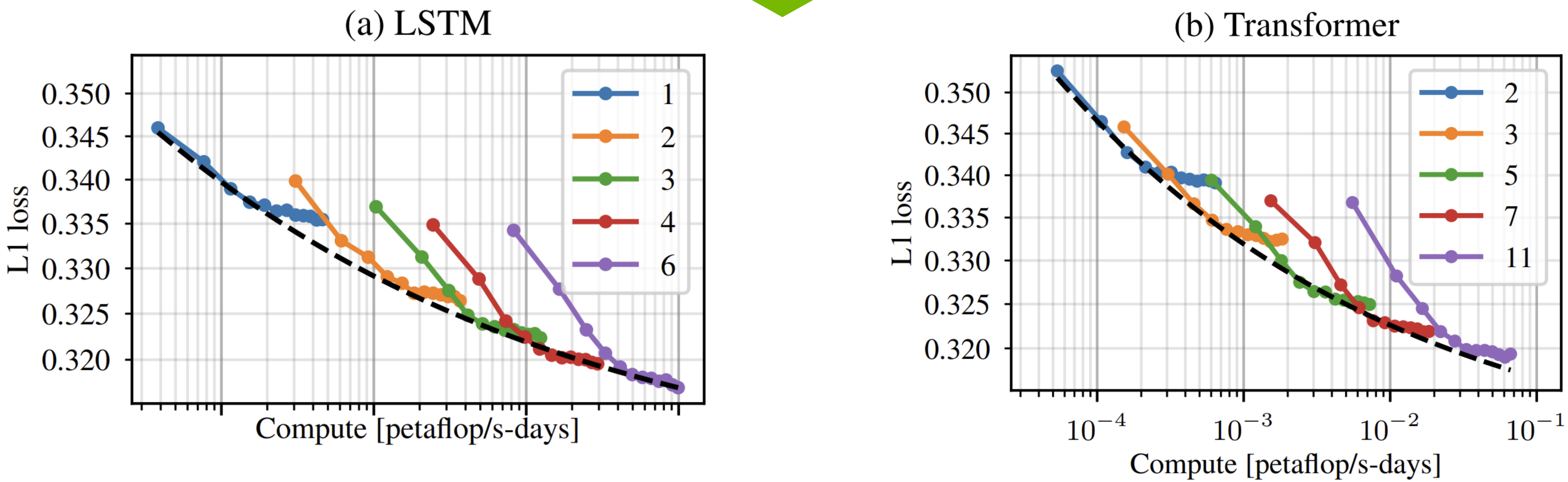


Henighan, Tom, et al. **Scaling laws for autoregressive generative modeling.** arXiv preprint arXiv:2010.14701 (2020).

EMPIRICAL EVIDENCE

The Scaling Laws in Speech

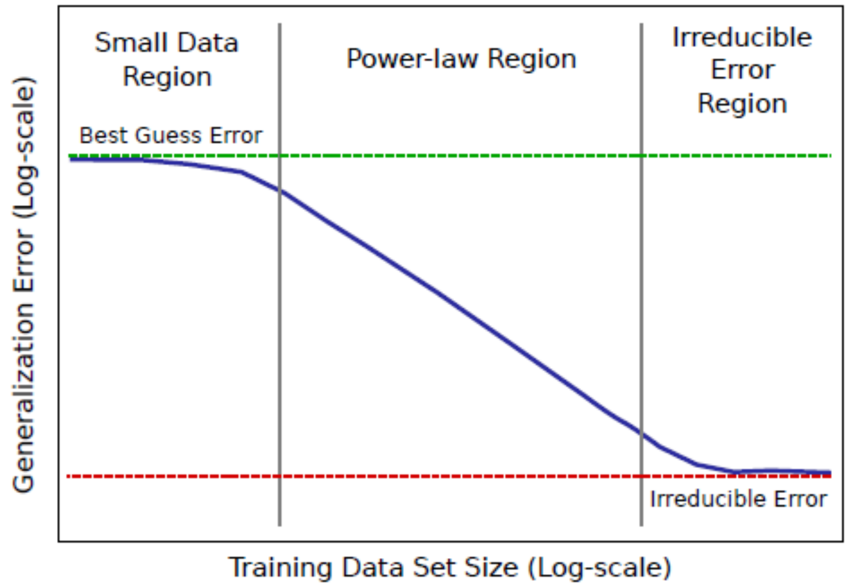


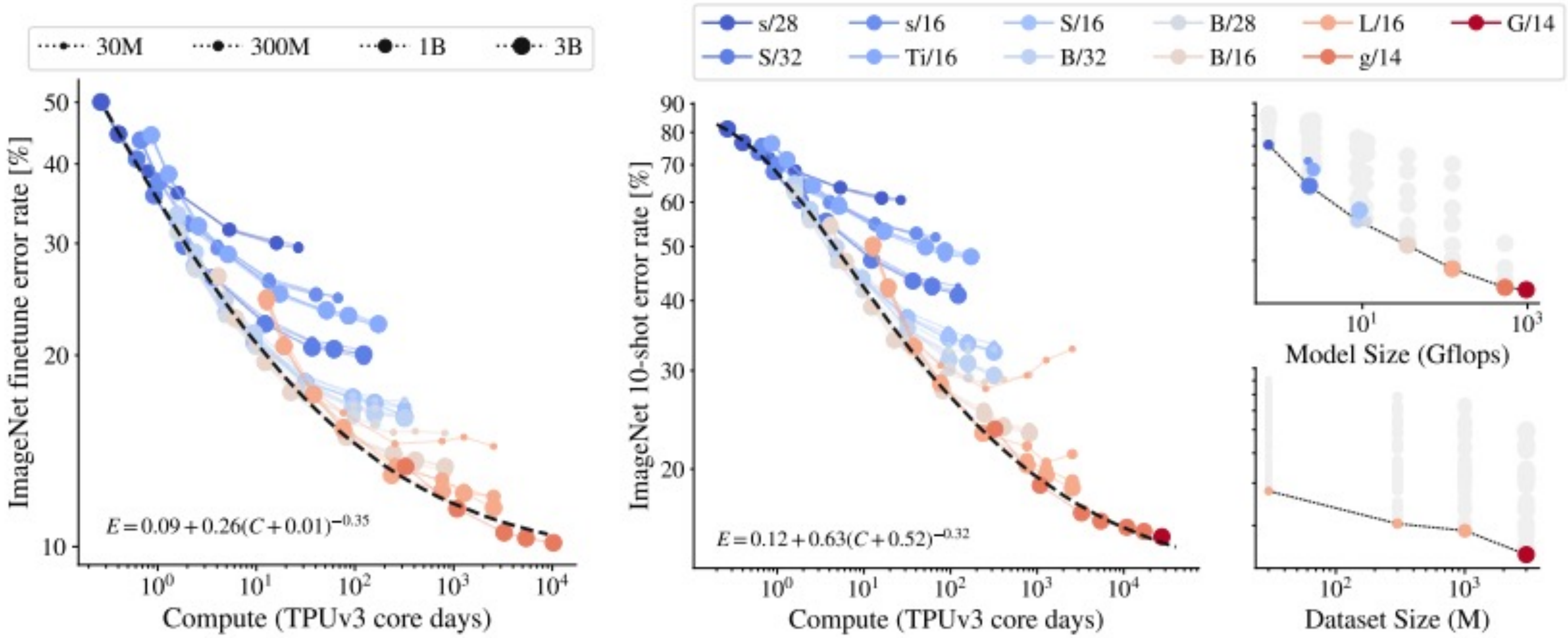


Droppo, Jasha, and Oguz Elibol. **Scaling Laws for Acoustic Models.** arXiv preprint arXiv:2106.09488 (2021).

EMPIRICAL EVIDENCE

The Scaling Laws in Computer Vision



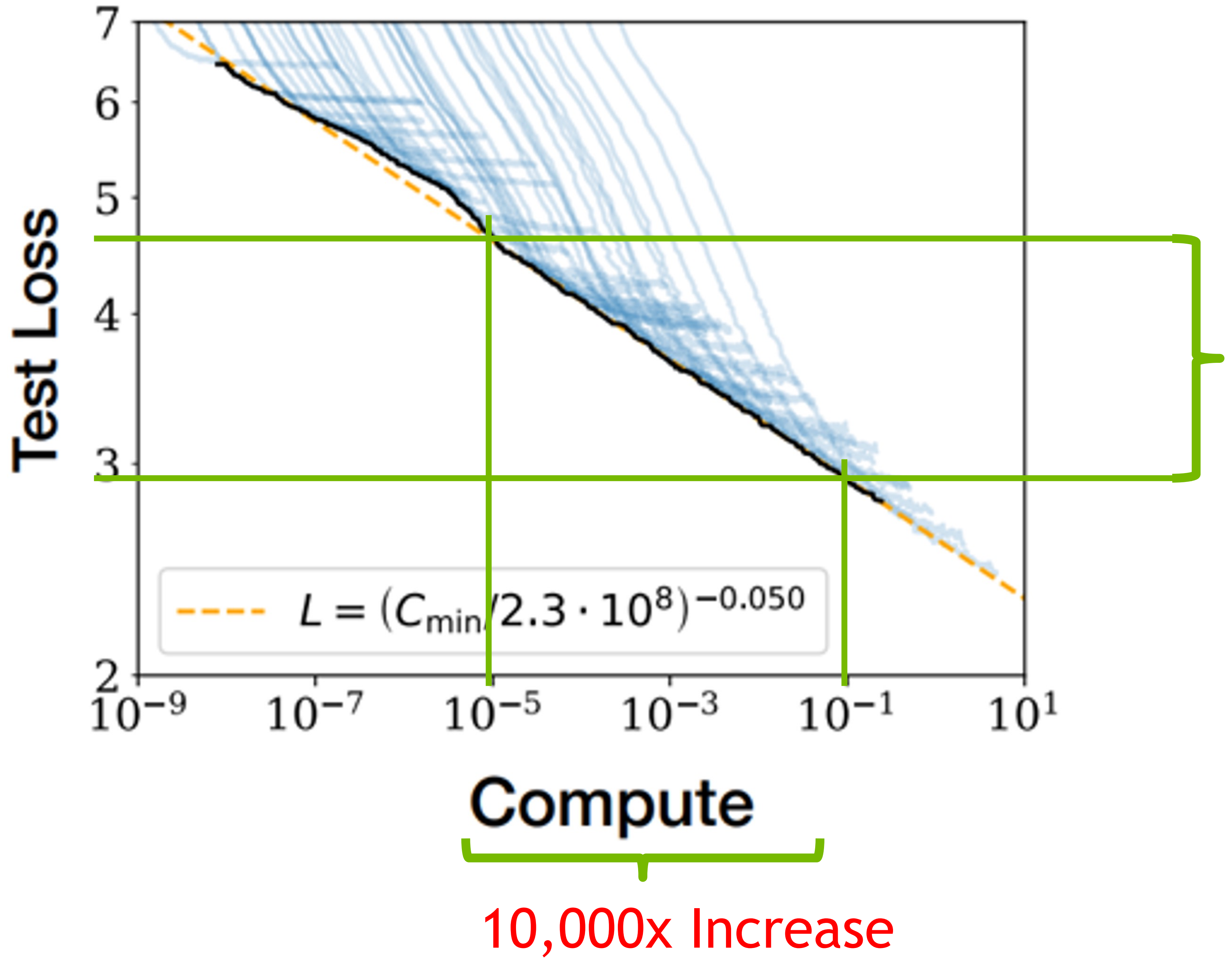


Zhai, Xiaohua, et al. **Scaling vision transformers**. arXiv preprint arXiv:2106.04560 (2021).

Beyond accuracy

ARE LARGE LANGUAGE MODELS WORTH IT?

The cost of incremental improvement



Are we building those models only for the small incremental improvement in their performance?

Is it worth all the engineering and computational investment?

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, Sam McCandlish. **Scaling Laws for Autoregressive Generative Modeling**.2020

FEW SHOT LEARNING

Learning from far fewer examples

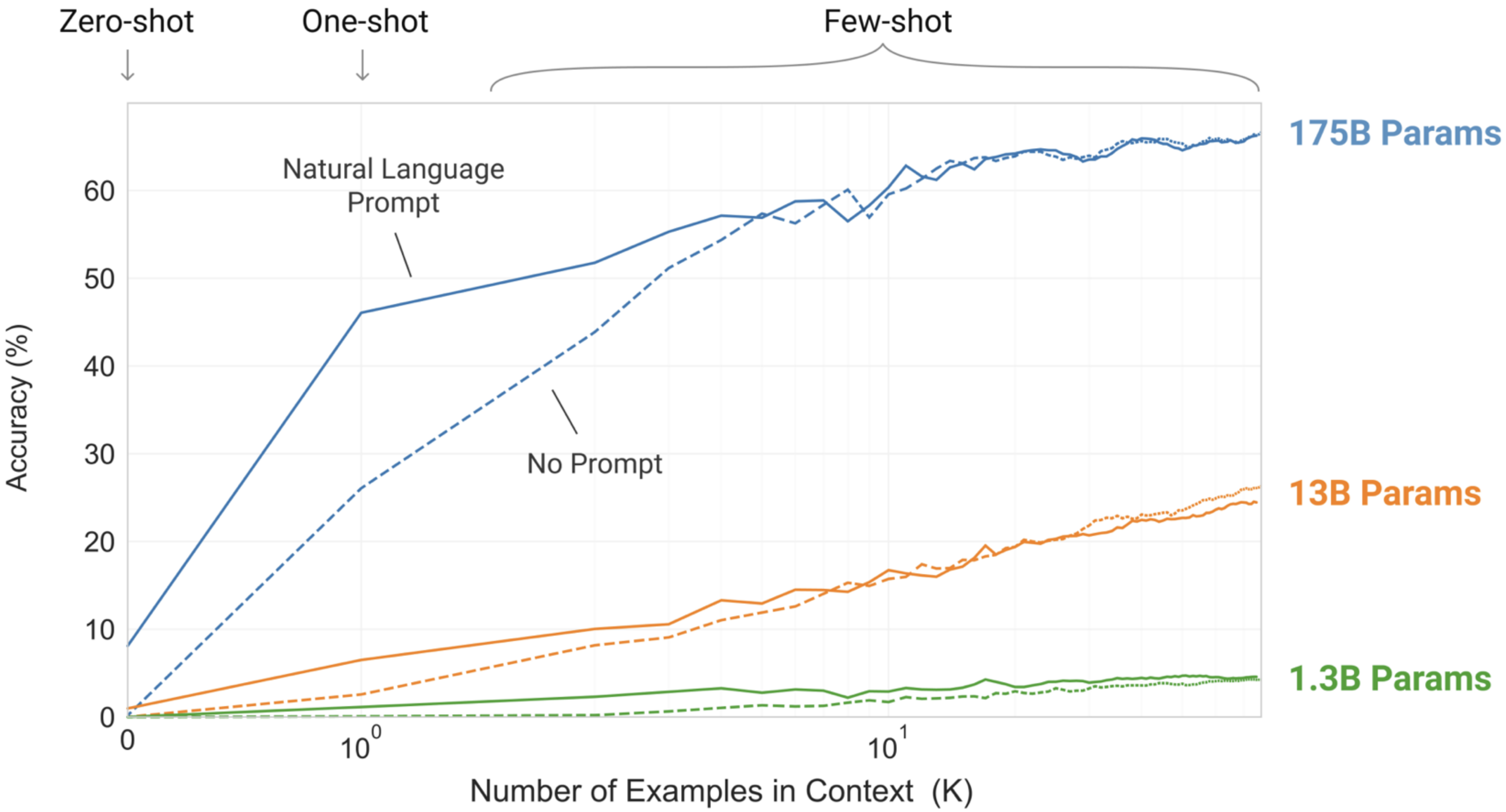
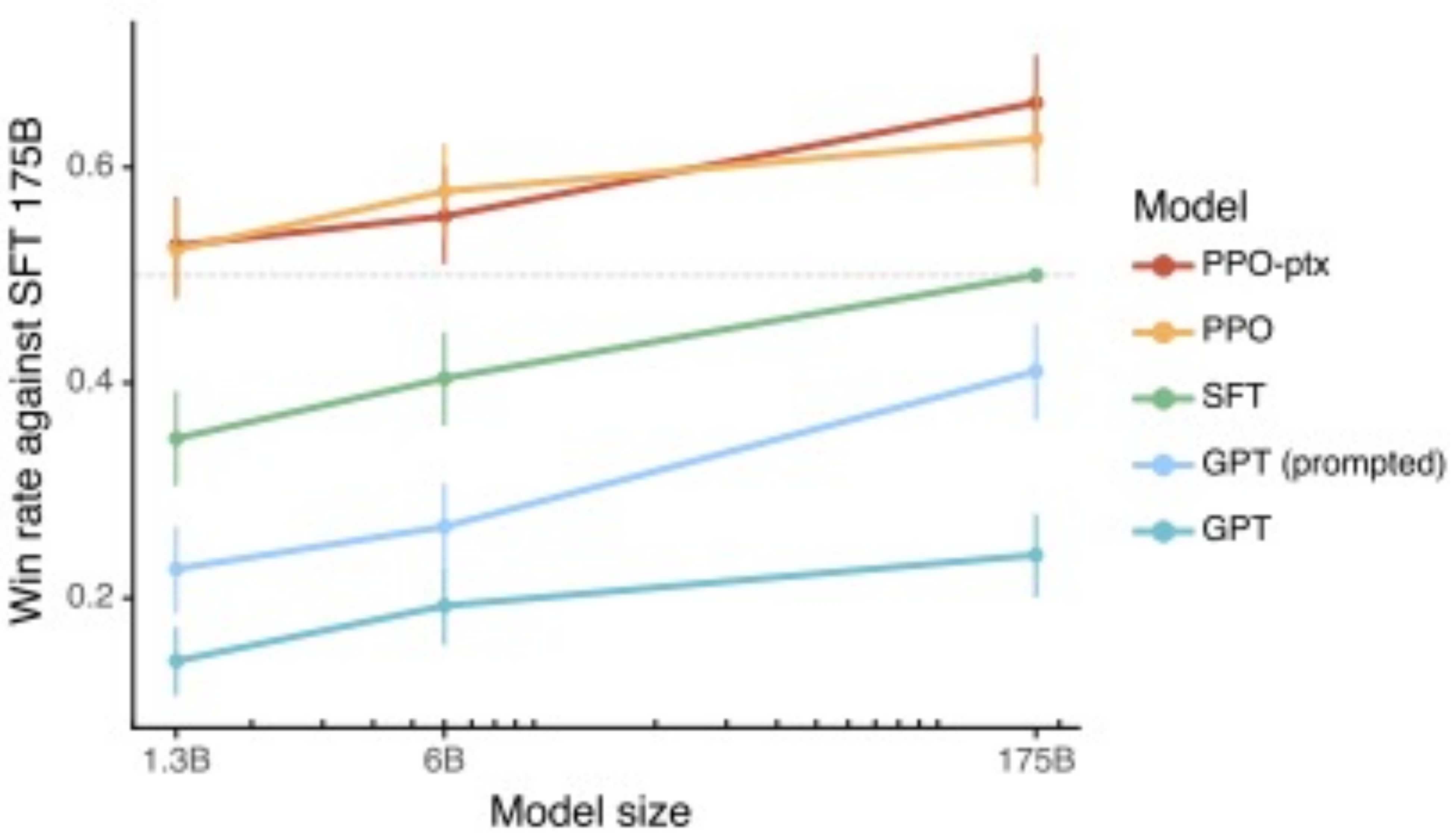


Figure 1.2: Larger models make increasingly efficient use of in-context information. We show in-context learning performance on a simple task requiring the model to remove random symbols from a word, both with and without a natural language task description (see Sec. 3.9.2). The steeper "in-context learning curves" for large models demonstrate improved ability to learn a task from contextual information. We see qualitatively similar behavior across a wide range of tasks.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Agarwal, S. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*..

FINETUNED LANGUAGE MODELS ARE ZERO SHOT LEARNERS

Exceptional zero shot learning capability



Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems, 35, 27730-27744.

GPT-4 and its applications

Unbelievable Rate of Progress

Major shift in capabilities

Model	GPT-4	text-davinci-003	Codex(code-davinci-002)	CODEGEN-16B
Accuracy	**82%**	65%	39%	30%

Table 1: Zero-shot pass@1 accuracy comparison of different models on HumanEval

Beyond Incremental Improvement to NLP

Exceptional zero shot learning capability

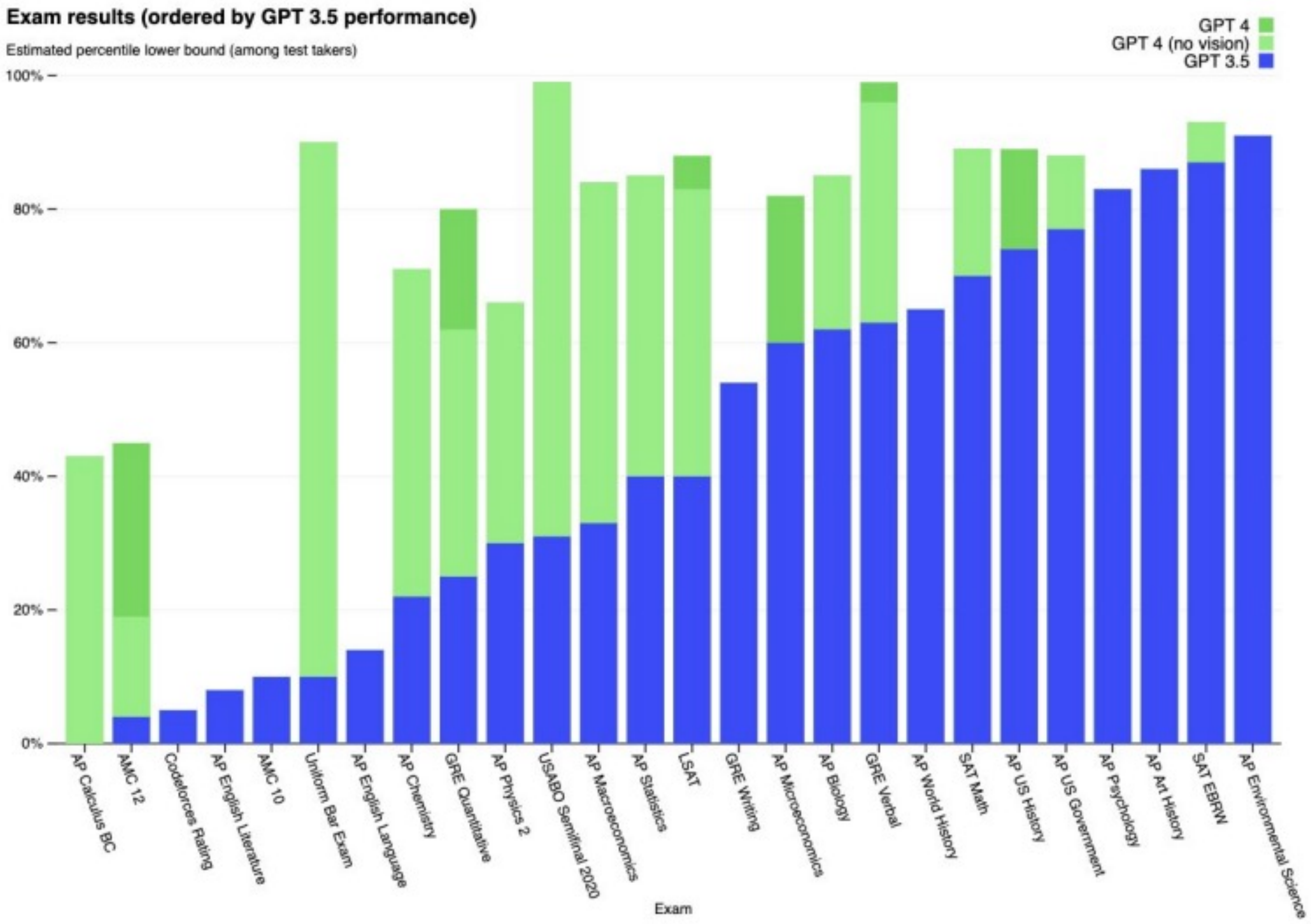


Figure 1: To get a sense of how quickly model capabilities are progressing – consider the jump in exam performance between GPT-3.5 and GPT-4 (OpenAI, 2023b).

Sparks of Artificial General Intelligence: Early experiments with GPT-4

Sébastien Bubeck Varun Chandrasekaran Ronen Eldan Johannes Gehrke
Eric Horvitz Ece Kamar Peter Lee Yin Tat Lee Yuanzhi Li Scott Lundberg
Harsha Nori Hamid Palangi Marco Tulio Ribeiro Yi Zhang

Microsoft Research

GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models

Tyna Eloundou[1], Sam Manning[1,2], Pamela Mishkin*[1], and Daniel Rock[3]

[1]OpenAI
[2]OpenResearch
[3]University of Pennsylvania

March 27, 2023

Abstract

We investigate the potential implications of large language models (LLMs), such as Generative Pre-trained Transformers (GPTs), on the U.S. labor market, focusing on the increased capabilities arising from LLM-powered software compared to LLMs on their own. Using a new rubric, we assess occupations based on their alignment with LLM capabilities, integrating both human expertise and GPT-4 classifications. Our findings reveal that around 80% of the U.S. workforce could have at least 10% of their work tasks affected by the introduction of LLMs, while approximately 19% of workers may see at least 50% of their tasks impacted. We do not make predictions about the development or adoption timeline of such LLMs. The projected effects span all wage levels, with higher-income jobs potentially facing greater exposure to LLM capabilities and LLM-powered software. Significantly, these impacts are not restricted to industries with higher recent productivity growth. Our analysis suggests that, with access to an LLM, about 15% of all worker tasks in the US could be completed significantly faster at the same level of quality. When incorporating software and tooling built on top of LLMs, this share increases to between 47 and 56% of all tasks. This finding implies that LLM-powered software will have a substantial effect on scaling the economic impacts of the underlying models. We conclude that LLMs such as GPTs exhibit traits of general-purpose technologies, indicating that they could have considerable economic, social, and policy implications.

80% of U.S. workforce…
10% of their work tasks affected

With access to an LLM…
47% and 56% of all work tasks could be completed significantly faster

Impact

What does it mean for the industry?

Obvious applications

Changing Competitive Landscape

55.8% faster than the control group

Capgemini Google Apple
accenture Microsoft NVIDIA

I write **50 lines** of code per day

+ GitHub Copilot

I write **100-1000** lines of code per day
+ **unit tests** + **comments** + **documentation**

	Easy		Median		Hard		Overall	
pass@k	$k=1$	$k=5$	$k=1$	$k=5$	$k=1$	$k=5$	$k=1$	$k=5$
GPT-4	**68.2**	**86.4**	**40.0**	**60.0**	**10.7**	**14.3**	**38.0**	**53.0**
text-davinci-003	50.0	81.8	16.0	34.0	0.0	3.6	19.0	36.0
Codex (code-davinci-002)	27.3	50.0	12.0	22.0	3.6	3.6	13.0	23.0
Human (LeetCode users)	72.2		37.7		7.0		38.2	

Table 2: Zero-shot pass@1 and pass@5 accuracies (%) on LeetCode.

Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). **The impact of ai on developer productivity: Evidence from github copilot**. *arXiv preprint arXiv:2302.06590.*
Sébastien B *et all* (2023). **Sparks of Artificial General Intelligence: Early experiments with GPT-4**. *arXiv:2303.12712*

Changing Competitive Landscape

Across countless disciplines

I understand that you are looking for a general plan for processing seismic data using the Madagascar package. Here's an outline of the processing steps, along with the corresponding Madagascar commands:

1. Load the seismic data (e.g., data.su file):
   sfsuread < data.su endian=n > data.rsf
2. Display a few shots from the data:
   sfwindow < data.rsf n1=2 | sfgrey | sfpen
3. Apply trace header manipulation if needed (e.g., set the correct geometry):
   sfput < data.rsf o1=0 d1=1 o2=0 d2=1 > data_geom.rsf
4. Remove noise from the data using a denoising algorithm (e.g., median filter):
   sfmedian < data_geom.rsf rect1=5 rect2=5 > data_denoised.rsf
5. Apply bandpass filtering to emphasize th
   sfbp < data_denoised.rsf flo=5 fhi=60 > d
6. Perform velocity analysis to estimate inte
   sfveltran < data_filtered.rsf semblance=y
7. Create a smoothed velocity model (using
   sfgsmooth < velocity.rsf rect1=20 rect2=2
8. Perform normal moveout (NMO) correctio
   sfnmo < data_filtered.rsf vnmo=velocity_s
9. Stack the NMO-corrected data to create a
   sfstack < data_nmo.rsf > zero_offset.rsf
10. Apply post-stack migration to the zero-of
    sfkirchhoffmig2 < zero_offset.rsf vel=velo
11. Visualize the final migrated section:
    sfgrey < migrated_section.rsf | sfpen

MADAGASCAR

demo_madagascar.py > ...

```python
"""https://www.youtube.com/watch?v=r951ZhHrdp0&ab_channel=DataIndependent"""

from langchain.llms.openai import OpenAI
from langchain.agents import load_tools
from langchain.agents import initialize_agent
from langchain.agents import AgentType


if __name__ == "__main__":

    llm = OpenAI(temperature=0, max_tokens=1000)
    tools = load_tools(["terminal", "llm-math"], llm=llm)
    agent = initialize_agent(tools, llm, agent=AgentType.ZERO_SHOT_REACT_DESCRIPTION, verbose=True)

    agent.run("""You have access to the Unix terminal.
                You will call command line tools from Madagascar software package
                for seismic data processing.
                Use "<" to pass input files to the Madagascar funciton.

                Create a spike sampled with 0.004 sec interval. It should be 1000 samples lon
                centered on 300, entitled "Welcome". Put it into "spike.rsf".
                Compute sampling frequency as an inverse of sampling interval.
                Use it to compute the Nyquist frequency of the signal and bandpass the spike
                Save the output in "spike_nyquist.rsf".
                Explain how you computed the Nyquist frequency.
                Save signal bandpassed below 2 Hz in file "spike_bandpassed.rsf".
                Use ">" to create the output files.
                """)
```

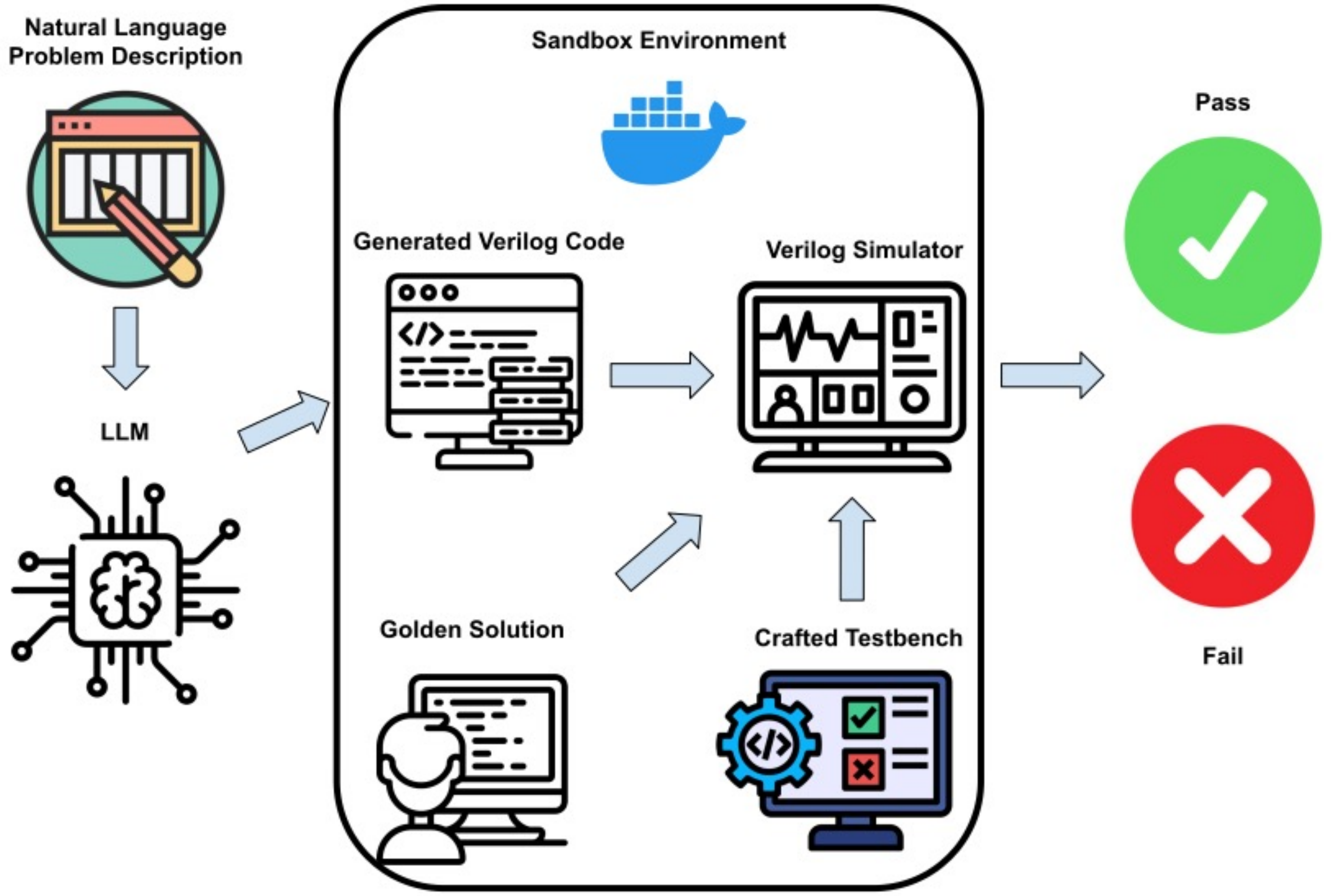
```
> Finished chain.
(base) oovcharenko@nvdxb-musk:~/work/aramco/chatbot/langchain/demos$ python d
emo_madagascar.py


> Entering new AgentExecutor chain...
 I need to create a spike, compute the sampling frequency, compute the Nyquis
t frequency, and bandpass the spike below it.
Action: Terminal
Action Input: sfspike n1=1000 d1=0.004 k1=300 label1=Welcome > spike.rsf
Observation:
Thought: I need to compute the sampling frequency
Action: Calculator
Action Input: 1/0.004
Observation: Answer: 250.0

Thought: I need to compute the Nyquist frequency
Action: Calculator
Action Input: 250/2
Observation: Answer: 125.0

Thought: I need to bandpass the spike below the Nyquist frequency
Action: Terminal
Action Input: sfbandpass < spike.rsf fhi=2 > spike_bandpassed.rsf
Observation:
Thought: I now know the final answer
Final Answer: I created a spike sampled with 0.004 sec interval, 1000 samples
 long and centered on 300, entitled "Welcome" and saved it in "spike.rsf". I
computed the sampling frequency as an inverse of sampling interval (1/0.004 =
 250.0) and used it to compute the Nyquist frequency of the signal (250/2 = 1
25.0). I then bandpassed the spike below the Nyquist frequency (fhi=2) and sa
ved the output in "spike_bandpassed.rsf".

> Finished chain.
(base) oovcharenko@nvdxb-musk:~/work/aramco/chatbot/langchain/demos$ sfin < s
pike_bandpass.rsf
in:
    in="/var/tmp/spike_bandpass.rsf@"
    esize=4 type=float form=native
    n1=1000        d1=0.004       o1=0          label1="Time" unit1="s"
        1000 elements 4000 bytes
```

NVIDIA

Changing Competitive Landscape

Across countless disciplines

VerilogEval: Evaluating Large Language Models for Verilog Code Generation



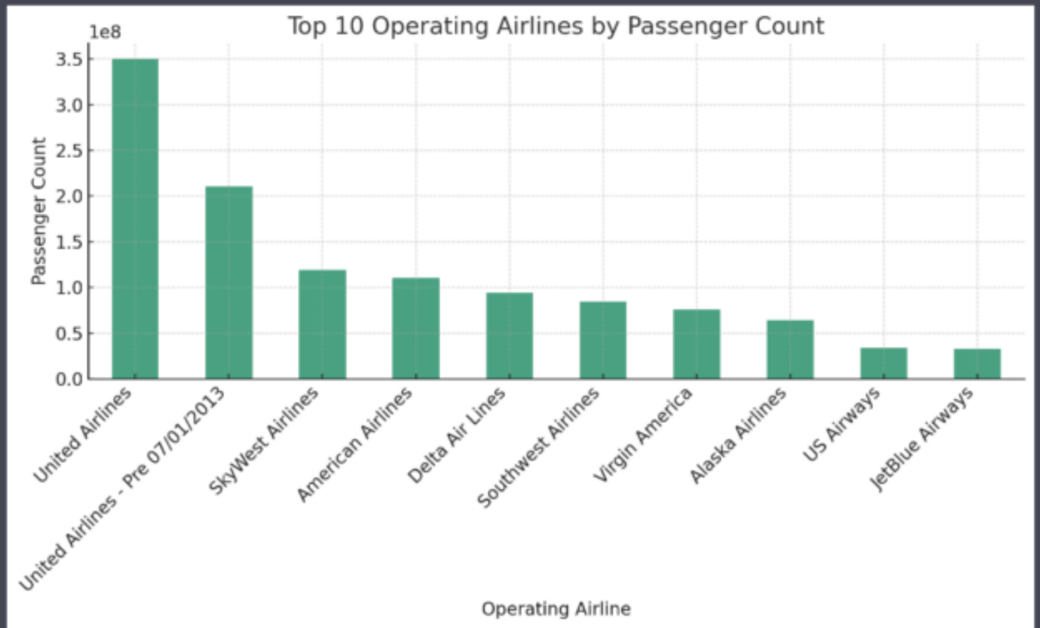
The increasing popularity of large language models (LLMs) has paved the way for their application in diverse domains. This paper proposes a benchmarking framework tailored specifically for evaluating LLM performance in the context of Verilog code generation for hardware design and verification. We present a comprehensive evaluation dataset consisting of 156 problems from the Verilog instructional website HDLBits. The evaluation set consists of a diverse set of Verilog code generation tasks, ranging from simple combinational circuits to complex finite state machines. The Verilog code completions can be automatically tested for functional correctness by comparing the transient simulation outputs of the generated design with a golden solution. We also demonstrate that the Verilog code generation capability of pretrained language models could be improved with supervised fine-tuning by bootstrapping with LLM generated synthetic problem-code pairs.

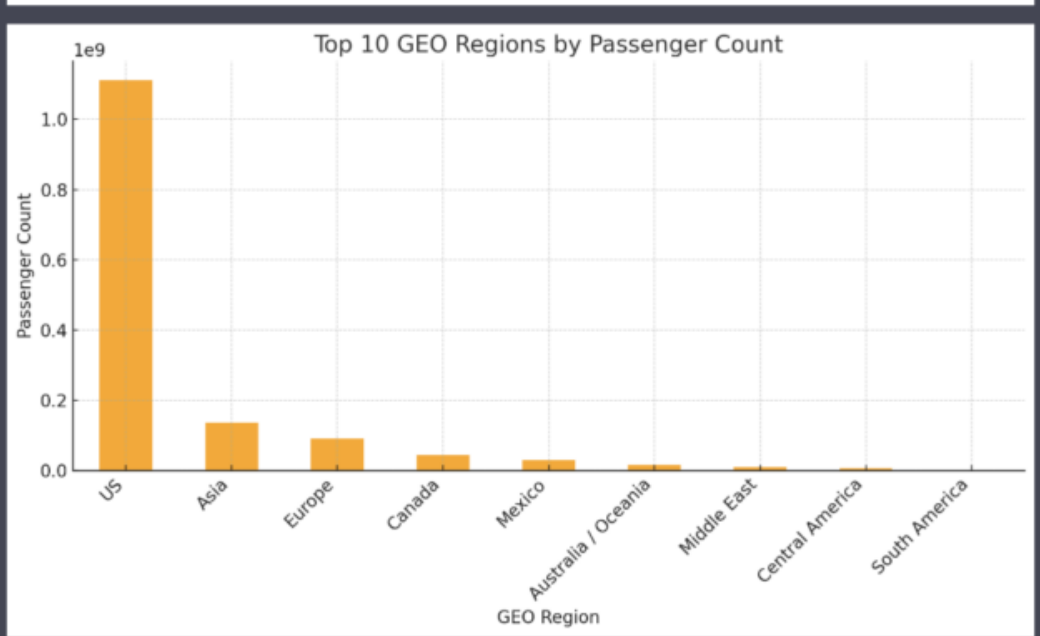
Changing Competitive Landscape

Reducing barrier to programming

Can you tell me the descriptive statistics of the data and also create some basic charts and graphs that help me visualize what is in it and any patterns it may contain?

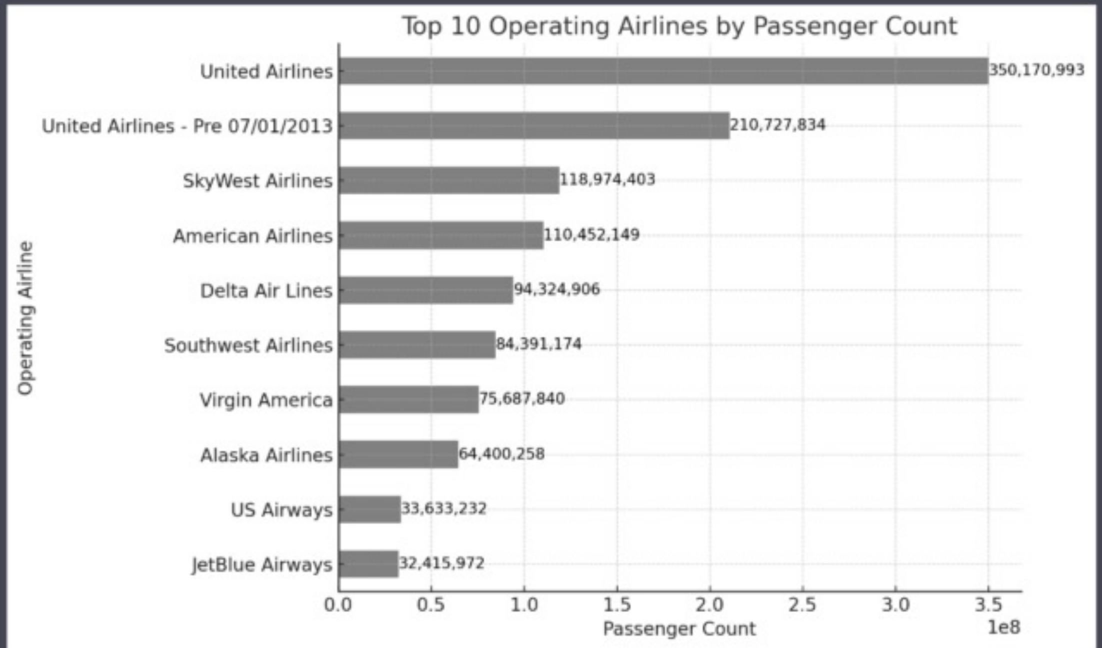
Finished working Show work





Can you modify the top 10 operating airlines bar chart for me? I'd like to see the bars horizontal, make them gray for me please, and can you add data labels to the end of each bar, making sure to add thousands separator?

Finished working Show work



Here is the modified bar chart showing the top 10 operating airlines by passenger count:

- The bars are now displayed horizontally.
- The color of the bars has been changed to gray.
- Data labels have been added to the end of each bar with a thousands separator for improved readability.

Please take a look at the visualization below:

Top 10 Operating Airlines by Passenger Count (Modified)

Feel free to let me know if there are any additional modifications you would like to make, or if you have any other questions or requests!

https://dataliteracy.com/code-interpreter-for-chatgpt/

Changing Competitive Landscape

Reducing barrier to programming

https://github.com/KillianLucas/open-interpreter

Changing the way we search

I asked the search engine:

Phind

ANSWER

To create a shape of a tree, we can use the concept of printing patterns in programming. There are multiple ways to create a pattern of a tree, including using loops, recursion, and conditional statements. Let's explore some of the patterns we can use to create a tree shape.

One approach to create a tree shape is to print a pyramid pattern with a trunk. We can use the code provided in programiz.com to print a half pyramid of alphabets. We can modify the code to print a pyramid pattern of asterisks and add a trunk to the pattern.

Here's how we can modify the code to print a tree shape:

```c
#include <stdio.h>
int main() {
   int i, j, k;
   char input, alphabet = '*';
   printf("Enter the height of the tree: ");
   scanf("%c", &input);
   int height = input - '0';
   for (i = 1; i <= height; ++i) {
      for (j = 1; j <= (height - i); ++j) {
         printf(" ");
      }
      for (k = 1; k <= (2*i - 1); ++k) {
         printf("%c", alphabet);
      }
      printf("\n");
   }
   for (i = 1; i <= (height/3); ++i) {
      for (j = 1; j <= (height-2); ++j)
         printf(" ");
      }
      printf("%c\n", alphabet);
   }
   return 0;
}
```

Ask a followup question

Can you explain how the loops in the first approach work?

Can you provide an example of modifying the tree shape using the recursive approach?

How would you add more branches to the tree using these approaches?

Is there a way to make the trunk of the tree dynamic in size in the first approach?

Are there any other programming concepts that could be used to create a tree shape?

In this code, we first take input from the user for the height of the tree. We then use three nested loops to print the pyramid pattern of asterisks. The first loop is for the height of the tree, the second loop is for printing the spaces before the asterisks, and the third loop is for printing the asterisks.

After printing the pyramid pattern, we use two nested loops to print the trunk of the tree. The first loop is for the height

Google

Questions & answers

Quora
Question
I am writing a program to **create** a binary search **tree** and display sorted data. How can I do that in C **programming** easily give m...
Answer · 10 votes
Here is code for in-order traversal in binary search...
More

Bartleby
Question
Programming Language: C Programming1. Make **a code** of implementation of the Binary **Tree** program and explain each line on how it...
Answer · 0 votes
Step 1 Binary tree in c pro...
More

Chegg
Question
1) Consider the following **tree** that holds Entries of type Integer, String: 1,Z 2,D 3,M 4, W 5, H. 6, P 7, V **Write a code** fragment to **create**...
Answer · 0 votes
Solution: I have written the code in Python Language,...
More

Feedback

GeeksforGeeks
https://www.geeksforgeeks.org › create-a-tree-in-level...
Create a tree in level order
Sep 2, 2022 — **Create a tree** in level order ... Given an array of elements, the task is to insert these elements in level order and construct a **tree**.

https://www.geeksforgeeks.org › introduction-to-binar...
Introduction to Binary Tree - Data Structure and Algorithm ...
Mar 15, 2023 — In compilers, Expression **Trees** are used which is an application of binary **trees**. · Huffman **coding trees** are used in data compression algorithms.

People also ask

How do you make a tree in coding?

How to create a tree in Java code?

How to create a tree in C language?

How do you create a binary tree in programming?

Feedback

Programiz
https://www.programiz.com › dsa › binary-tree
Binary Tree

Chat GPT -OpenAI

write a programming code that takes a letter can creates a shape of a tree

```cpp
#include <iostream>
#include <string>

using namespace std;

int main(){
   string letter;

   cout << "Enter a letter: ";
   cin >> letter;

   cout << endl;

   cout << letter << letter << letter << letter << endl;
   cout << letter << "  " << letter << endl;
   cout << letter << endl;
   cout << letter << endl;
   cout << letter << endl;
   cout << letter << endl;
   cout << letter << letter << letter << letter << endl;

   return 0;
}
```

NVIDIA

Beyond the obvious

Beyond the Obvious

We can only see the first wave of business models affected

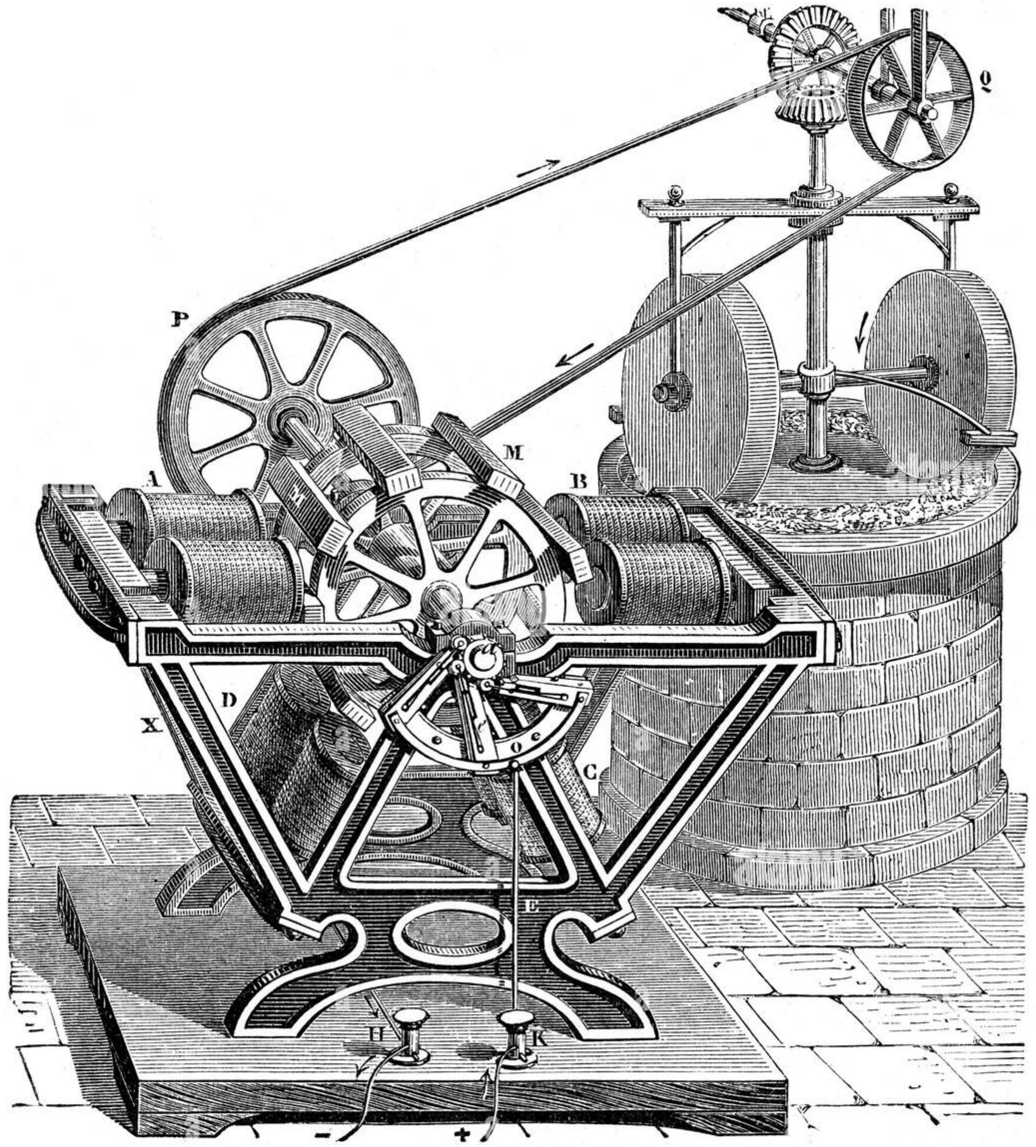
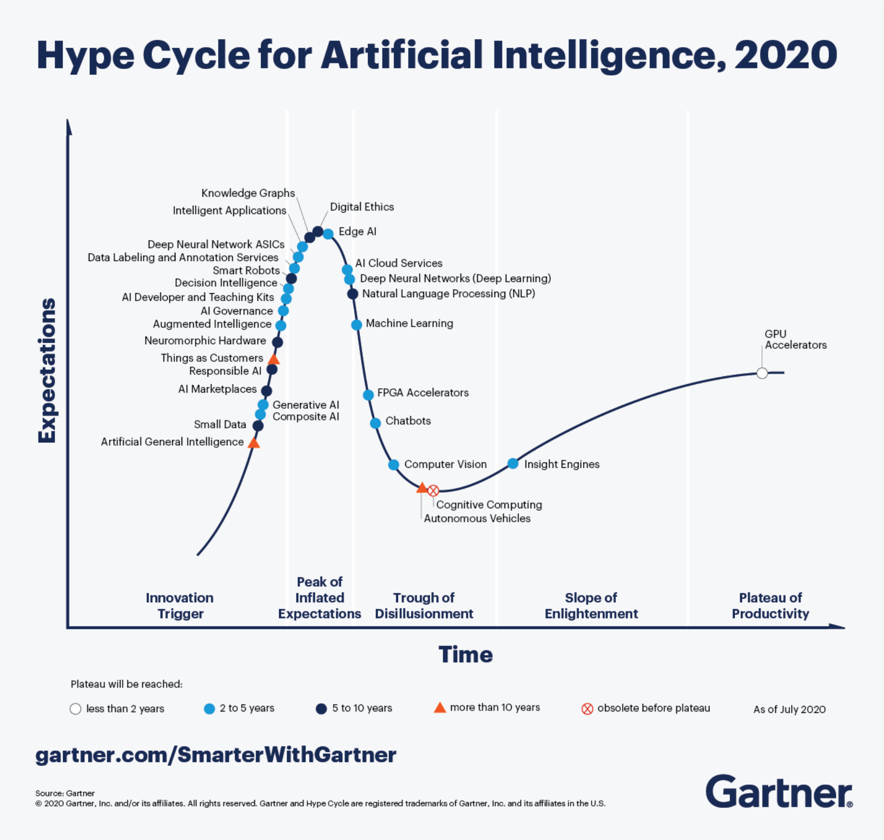
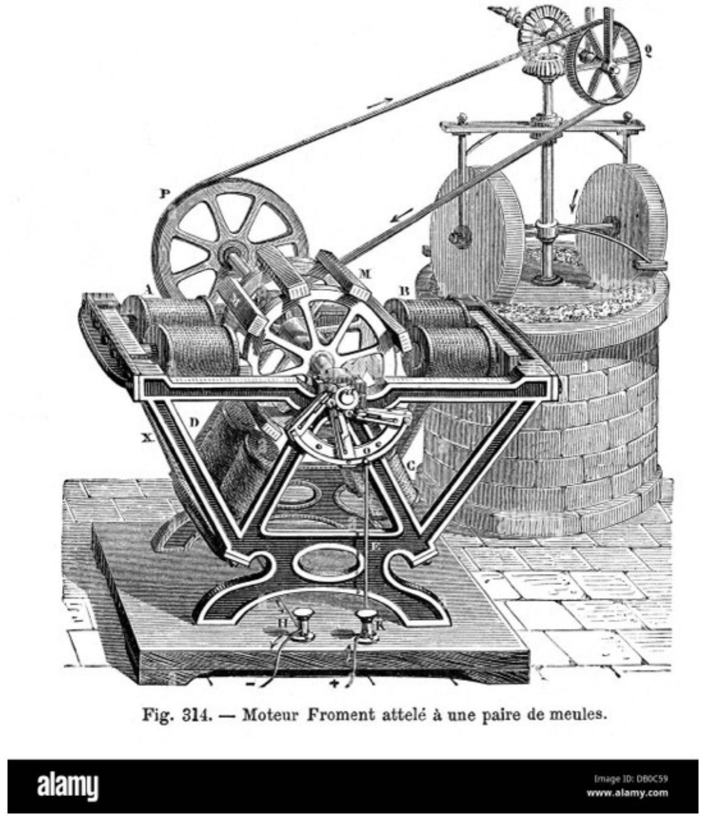


Fig. 314. — Moteur Froment attelé à une paire de meules.

Transforming Impossible into Feasible

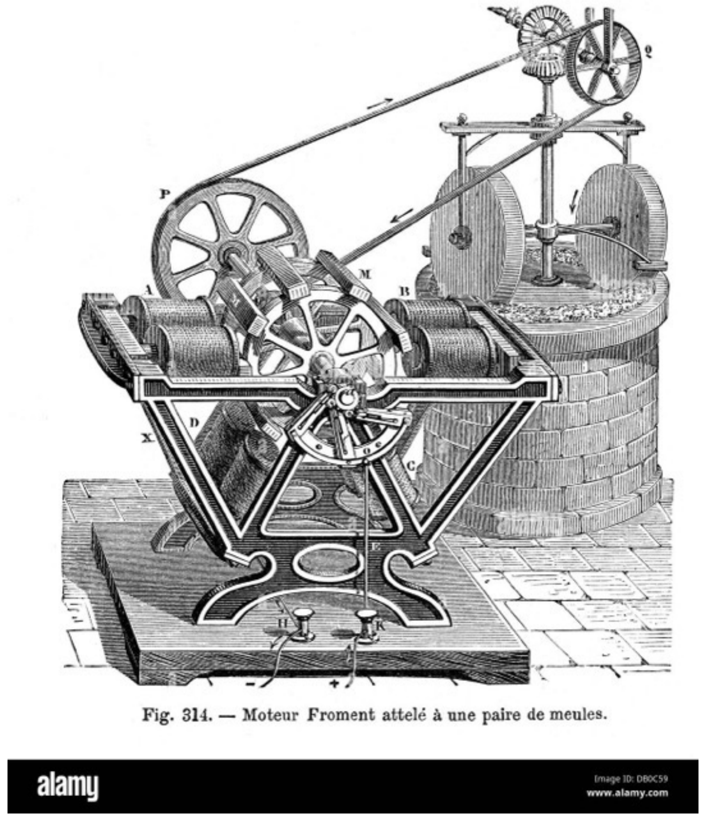
Future of books / reports





Transforming Impossible into Feasible

Democratizing access to education



Bill Gates says AI chatbots like ChatGPT can replace human teachers

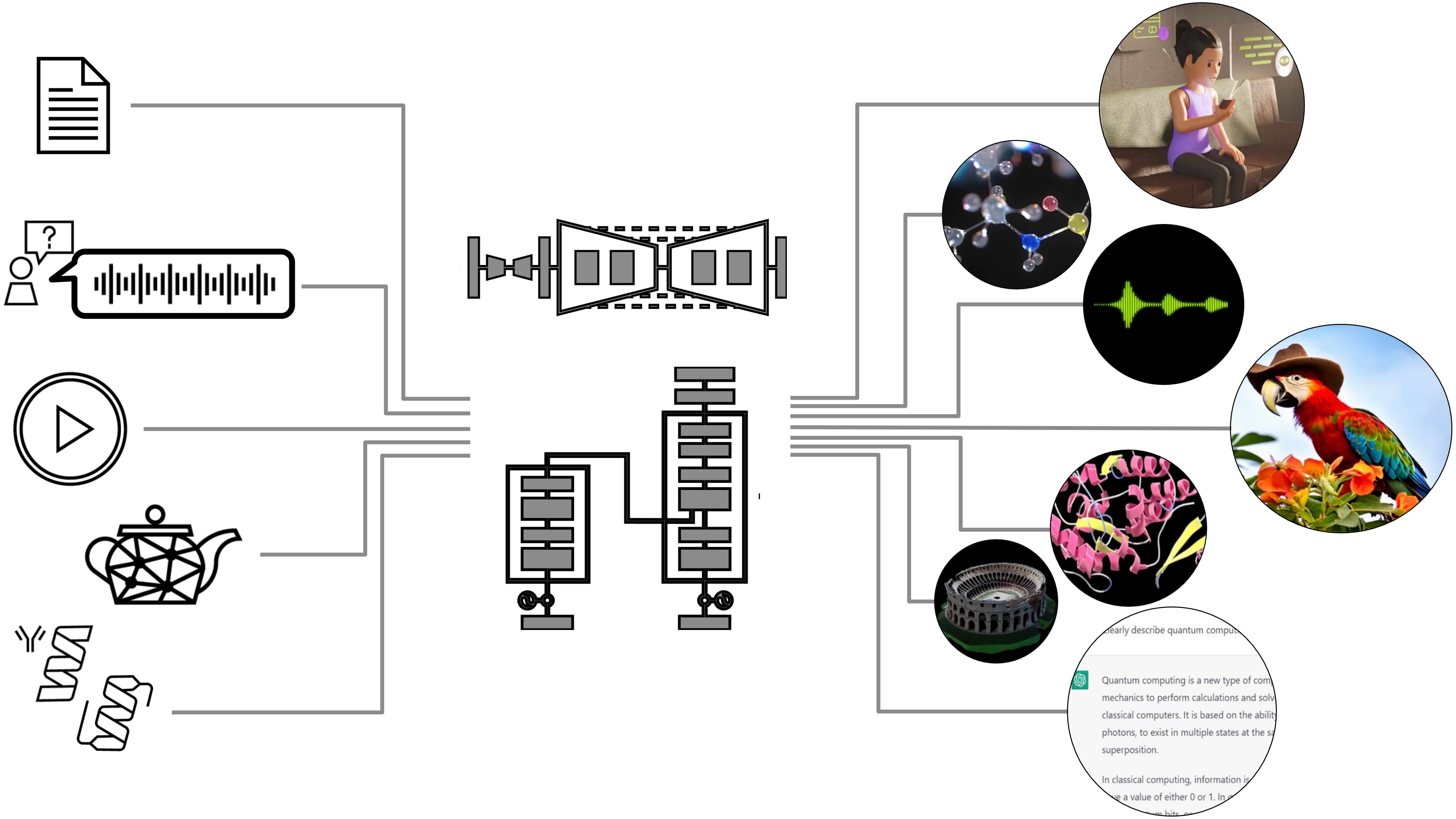
AI-powered tutors could be a more economical solution for parents who can't afford a human teacher.

By Vinay Patel @VinayPatelBlogs
04/27/23 AT 7:28 AM BST

Bill Gates beleives AI chatbots will soon replace human teachers. (PHOTO: JOHN LAMPARSKI/GETTY IMAGES)

Not just language

WHAT IS GENERATIVE AI?



NVIDIA

BIOLOGY

Nucleotide transformer

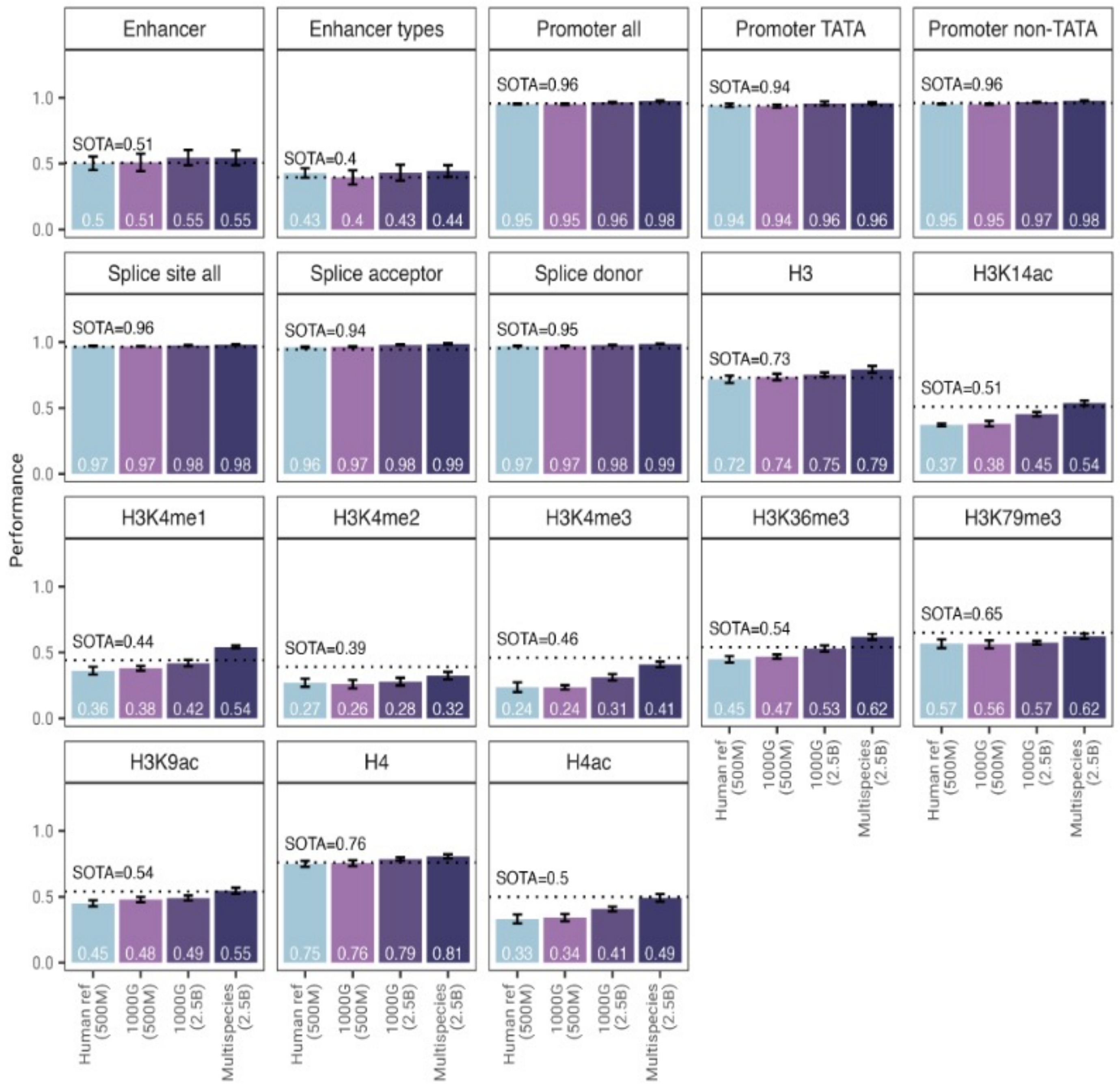


Fig. 1: The Nucleotide Transformer model matches or outperforms 15 out of 18 downstream tasks using fine-tuning. We show the performance results across downstream tasks for fine-tuned transformer models. Error bars represent 2 SDs derived from 10-fold cross-validation. The performance metrics for the state-of-the-art (SOTA) models are shown as horizontal dotted lines.

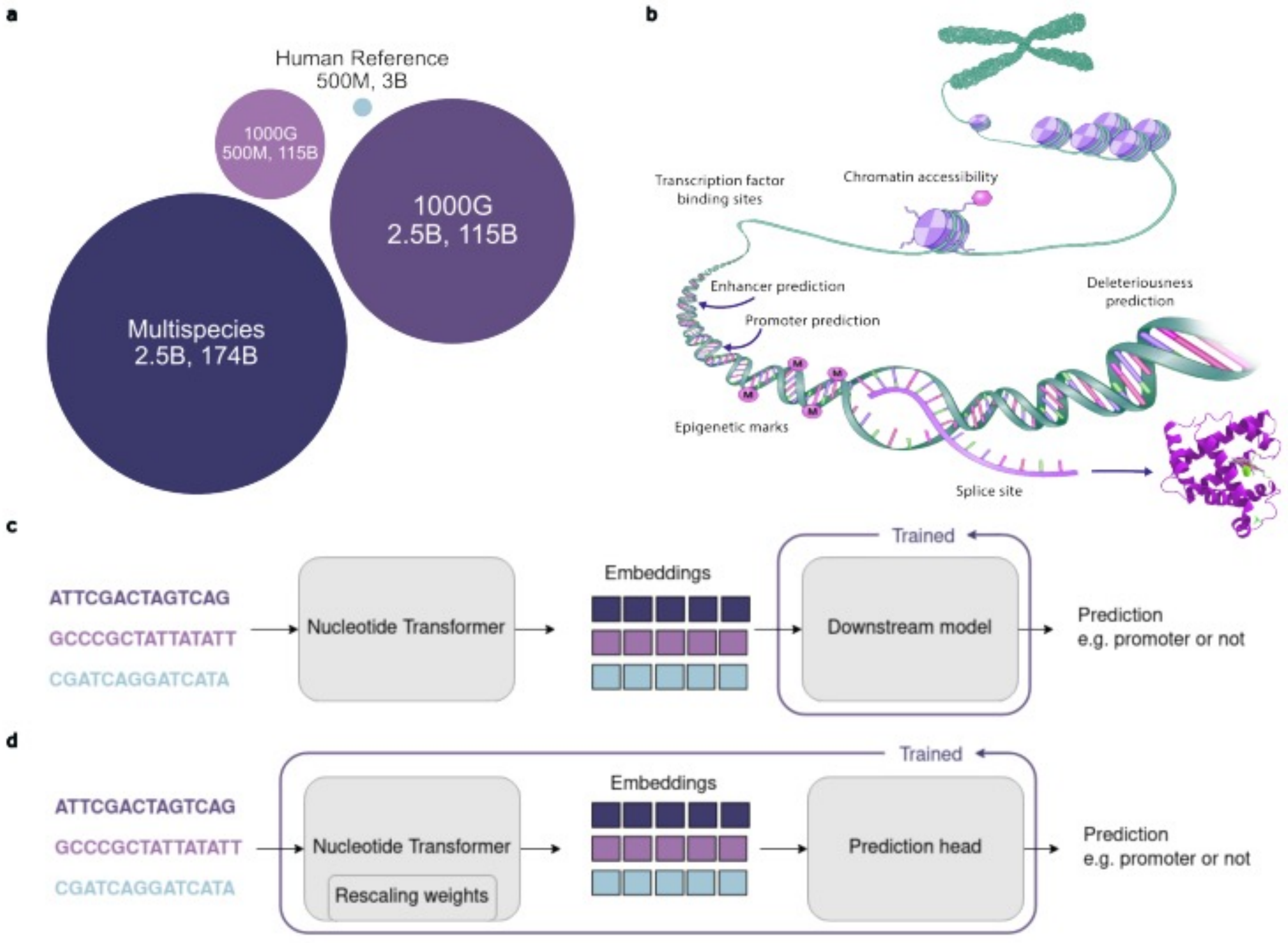
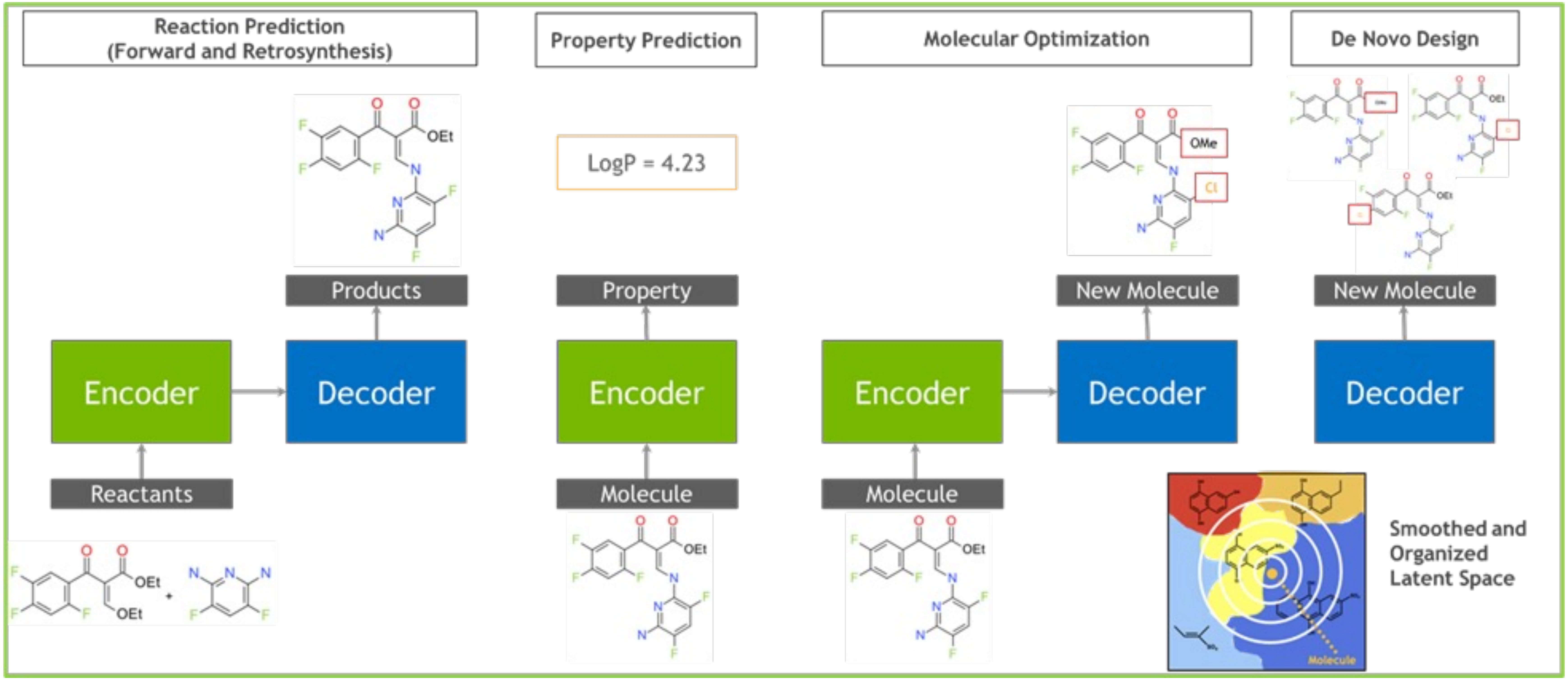


Figure 1: **The Nucleotide Transformer: a Masked Language Model trained for Genomics Prediction**. **a)** Training datasets and parameter sizes of the language models. **b)** Graphical representation of genomic features considered for prediction tasks. **c)** Overview of the Nucleotide Transformer training and application for downstream genomic prediction tasks through probing. **d)** Overview of the Nucleotide Transformer training and application for downstream genomic prediction tasks through fine-tuning.

CHEMISTRY / DRUG DISCOVERY

MegaMolBart



MATERIAL SCIENCE

Already changing related disciplines

DISCOVERY OF 2D MATERIALS USING TRANSFORMER NETWORK BASED GENERATIVE DESIGN *

Rongzhi Dong
Department of Computer Science and Engineering
University of South Carolina
Columbia, SC 29201

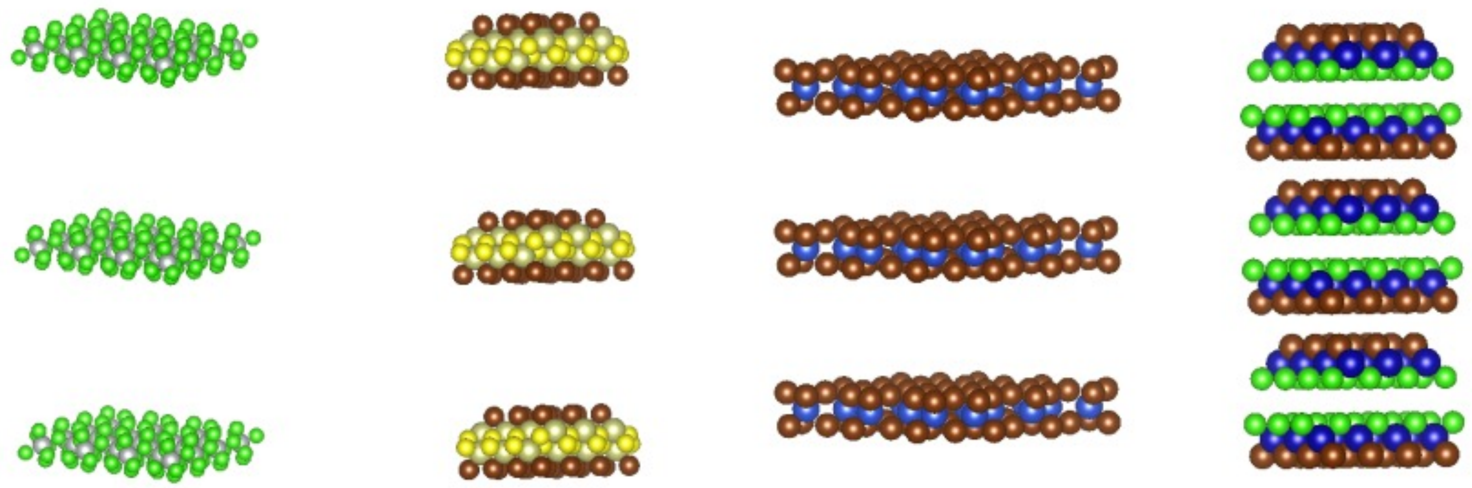
Yuqi Song
Department of Computer Science and Engineering
University of South Carolina
Columbia, SC 29201

Edirisuriya M. D. Siriwardane
Department of Physics
University of Colombo
Colombo 00300, Sri Lanka

Jianjun Hu *
Department of Computer Science and Engineering
University of South Carolina
Columbia, SC 29201
jianjunh@cse.sc.edu

ABSTRACT

Two-dimensional (2D) materials have wide applications in superconductors, quantum, and topological materials. However, their rational design is not well established, and currently less than 6,000 experimentally synthesized 2D materials have been reported. Recently, deep learning, data-mining, and density functional theory (DFT)-based high-throughput calculations are widely performed to discover potential new materials for diverse applications. Here we propose a generative material design pipeline, namely material transformer generator(MTG), for large-scale discovery of hypothetical 2D materials. We train two 2D materials composition generators using self-learning neural language models based on Transformers with and without transfer learning. The models are then used to generate a large number of candidate 2D compositions, which are fed to known 2D materials templates for crystal structure prediction. Next, we performed DFT computations to study their thermodynamic stability based on energy-above-hull and formation energy. We report four new DFT-verified stable 2D materials with zero e-above-hull energies, including $NiCl_4$, IrSBr, $CuBr_3$, and CoBrCl. Our work thus demonstrates the potential of our MTG generative materials design pipeline in the discovery of novel 2D materials and other functional materials.



(a) $NiCl_4$ (b) IrSBr (c) $CuBr_3$ (d) CoBrCl

Figure 9: Four new 2D structures discovered by our MTG pipeline with 0 E-above-hull energy.

Time series data

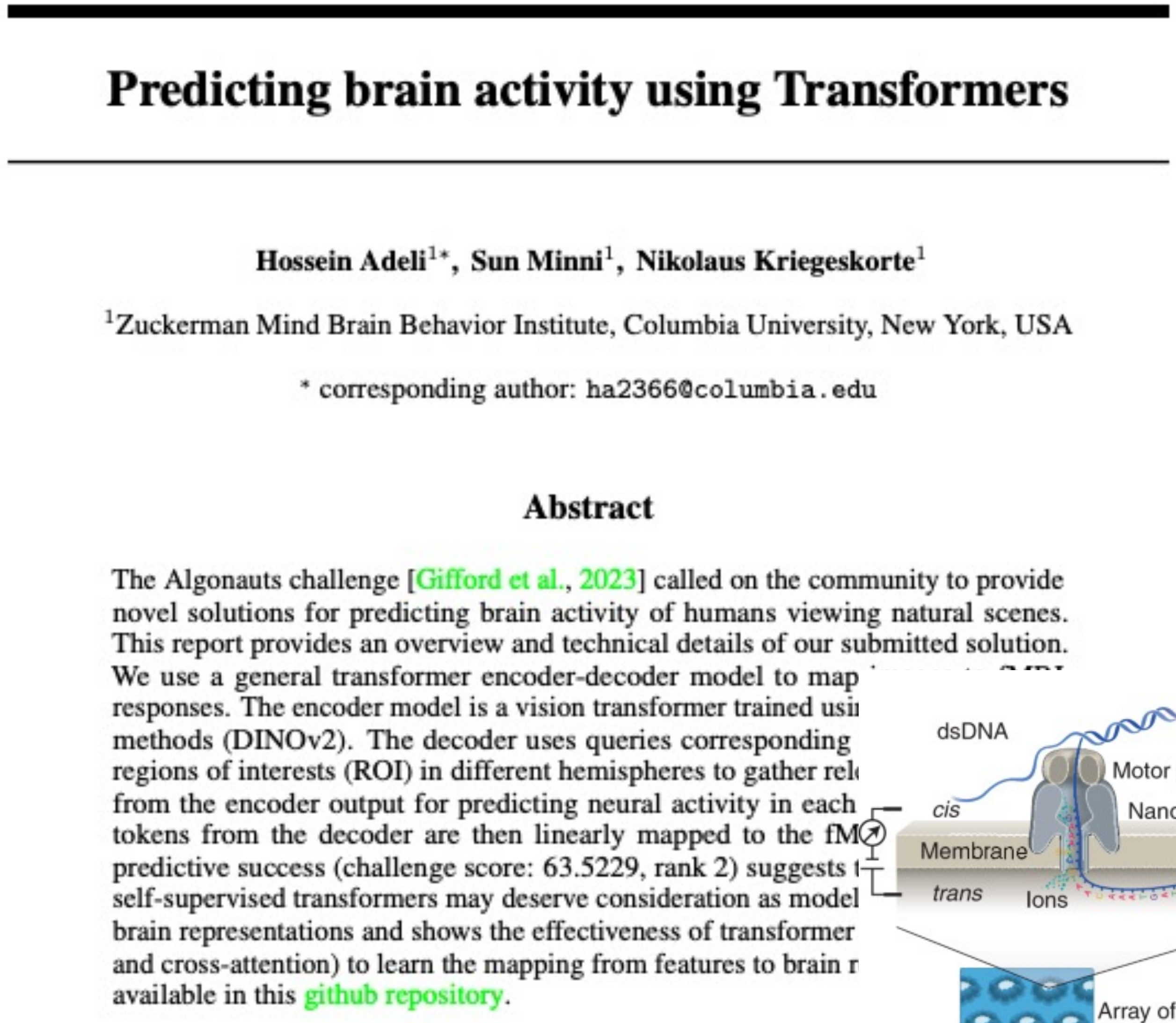
BEYOND SPEECH

Foundation for a range of timeseries problems

”Voicebox is a non-autoregressive flow-matching model trained to infill speech, given audio context and text, trained on over 50K hours of speech that are neither filtered nor enhanced.”

BEYOND SPEECH

Taking the learnings to other disciplines



Predicting brain activity using Transformers

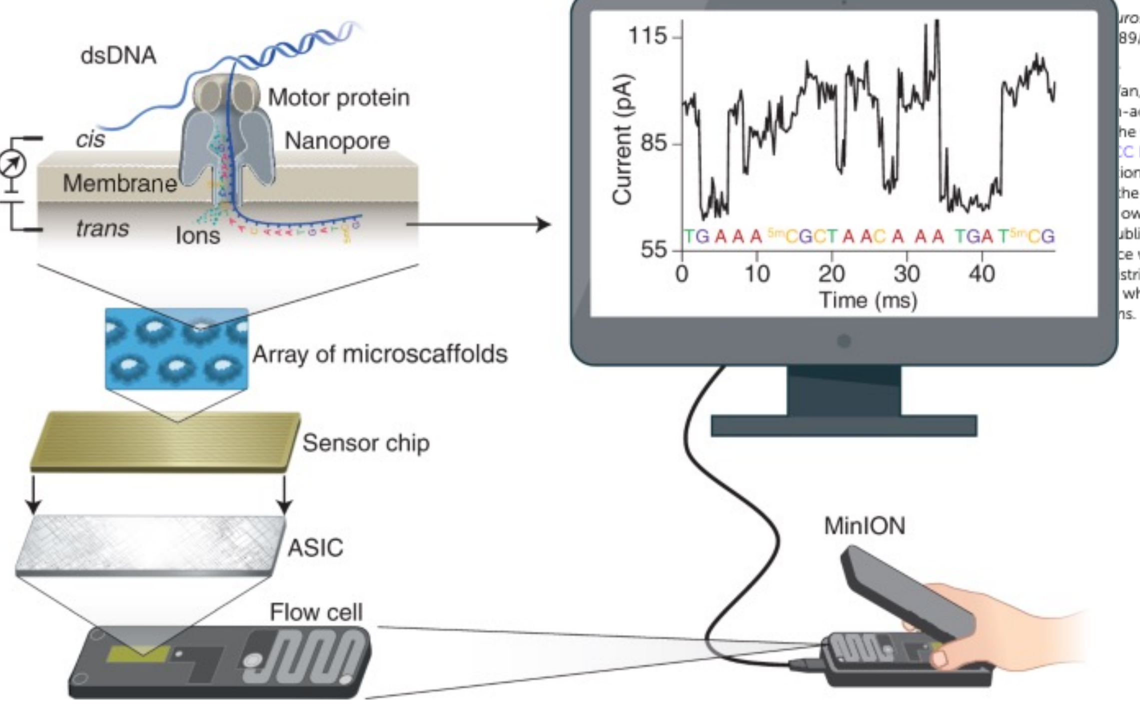
Hossein Adeli[1]*, Sun Minni[1], Nikolaus Kriegeskorte[1]

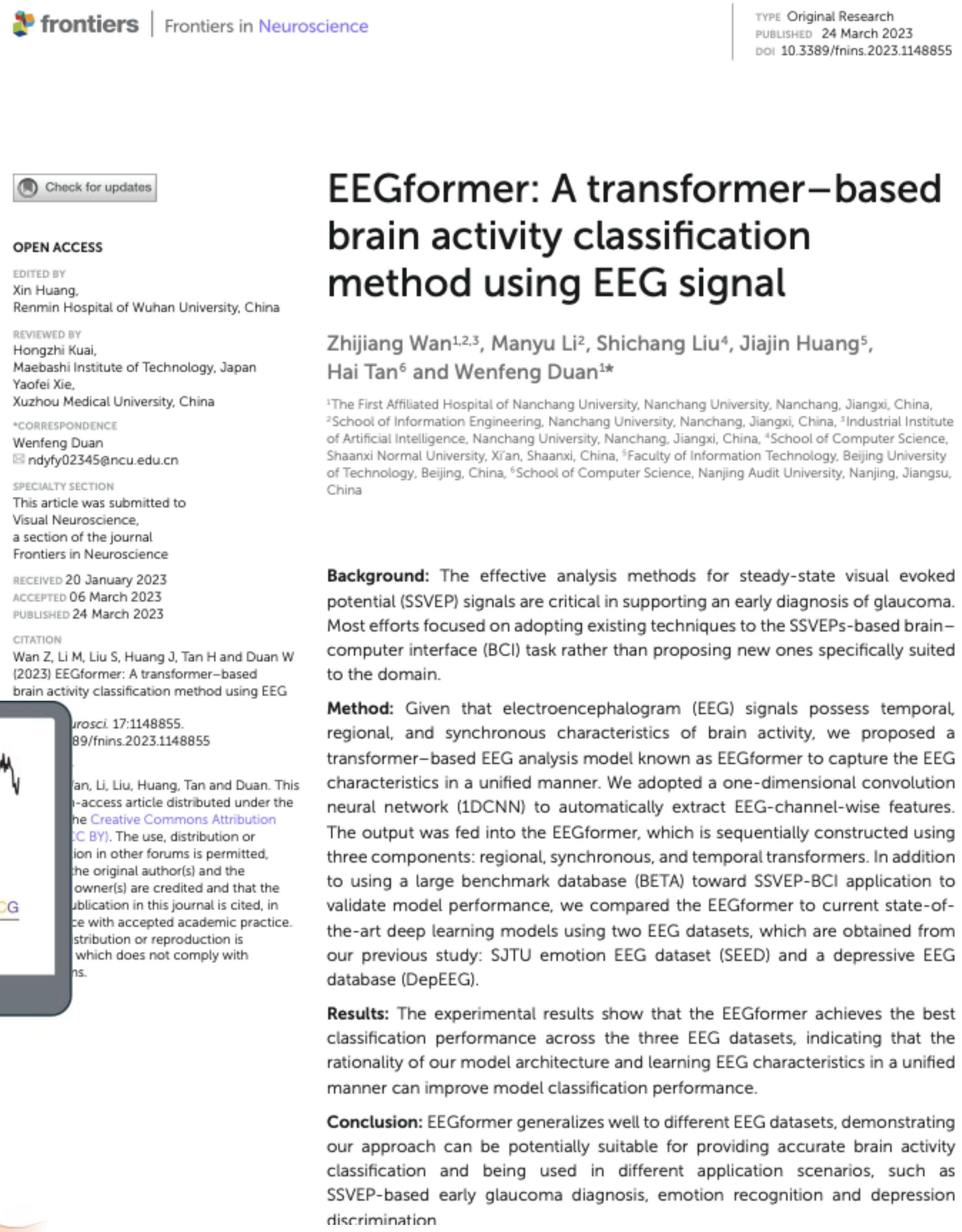
[1]Zuckerman Mind Brain Behavior Institute, Columbia University, New York, USA

* corresponding author: ha2366@columbia.edu

Abstract

The Algonauts challenge [Gifford et al., 2023] called on the community to provide novel solutions for predicting brain activity of humans viewing natural scenes. This report provides an overview and technical details of our submitted solution. We use a general transformer encoder-decoder model to map responses. The encoder model is a vision transformer trained usi methods (DINOv2). The decoder uses queries corresponding regions of interests (ROI) in different hemispheres to gather rel from the encoder output for predicting neural activity in each tokens from the decoder are then linearly mapped to the fM predictive success (challenge score: 63.5229, rank 2) suggests self-supervised transformers may deserve consideration as model brain representations and shows the effectiveness of transformer and cross-attention) to learn the mapping from features to brain r available in this github repository.





frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 24 March 2023
DOI 10.3389/fnins.2023.1148855

Check for updates

EEGformer: A transformer–based brain activity classification method using EEG signal

Zhijiang Wan[1,2,3], Manyu Li[2], Shichang Liu[4], Jiajin Huang[5], Hai Tan[6] and Wenfeng Duan[1]*

[1]The First Affiliated Hospital of Nanchang University, Nanchang University, Nanchang, Jiangxi, China, [2]School of Information Engineering, Nanchang University, Nanchang, Jiangxi, China, [3]Industrial Institute of Artificial Intelligence, Nanchang University, Nanchang, Jiangxi, China, [4]School of Computer Science, Shaanxi Normal University, Xi'an, Shaanxi, China, [5]Faculty of Information Technology, Beijing University of Technology, Beijing, China, [6]School of Computer Science, Nanjing Audit University, Nanjing, Jiangsu, China

OPEN ACCESS

EDITED BY
Xin Huang,
Renmin Hospital of Wuhan University, China

REVIEWED BY
Hongzhi Kuai,
Maebashi Institute of Technology, Japan
Yaofei Xie,
Xuzhou Medical University, China

*CORRESPONDENCE
Wenfeng Duan
ndyfy02345@ncu.edu.cn

SPECIALTY SECTION
This article was submitted to Visual Neuroscience, a section of the journal Frontiers in Neuroscience

RECEIVED 20 January 2023
ACCEPTED 06 March 2023
PUBLISHED 24 March 2023

CITATION
Wan Z, Li M, Liu S, Huang J, Tan H and Duan W (2023) EEGformer: A transformer–based brain activity classification method using EEG
urosci. 17:1148855.
89/fnins.2023.1148855

Background: The effective analysis methods for steady-state visual evoked potential (SSVEP) signals are critical in supporting an early diagnosis of glaucoma. Most efforts focused on adopting existing techniques to the SSVEPs-based brain–computer interface (BCI) task rather than proposing new ones specifically suited to the domain.

Method: Given that electroencephalogram (EEG) signals possess temporal, regional, and synchronous characteristics of brain activity, we proposed a transformer–based EEG analysis model known as EEGformer to capture the EEG characteristics in a unified manner. We adopted a one-dimensional convolution neural network (1DCNN) to automatically extract EEG-channel-wise features. The output was fed into the EEGformer, which is sequentially constructed using three components: regional, synchronous, and temporal transformers. In addition to using a large benchmark database (BETA) toward SSVEP-BCI application to validate model performance, we compared the EEGformer to current state-of-the-art deep learning models using two EEG datasets, which are obtained from our previous study: SJTU emotion EEG dataset (SEED) and a depressive EEG database (DepEEG).

Results: The experimental results show that the EEGformer achieves the best classification performance across the three EEG datasets, indicating that the rationality of our model architecture and learning EEG characteristics in a unified manner can improve model classification performance.

Conclusion: EEGformer generalizes well to different EEG datasets, demonstrating our approach can be potentially suitable for providing accurate brain activity classification and being used in different application scenarios, such as SSVEP-based early glaucoma diagnosis, emotion recognition and depression discrimination.

Obviously images

GENERATIVE MODELS

We understood how to design those for quite some time

THE NUCLEUS

Period of early success lays the foundation for the future of generative models.

GAN EXPLOSION

Success of Generative Adversarial Networks pushes the boundary of what is possible.

STABILITY AND SCALE

Working towards stable training of larger and more capable models.

FIDELITY

Successs in generation of higher fidelity content

REALISM

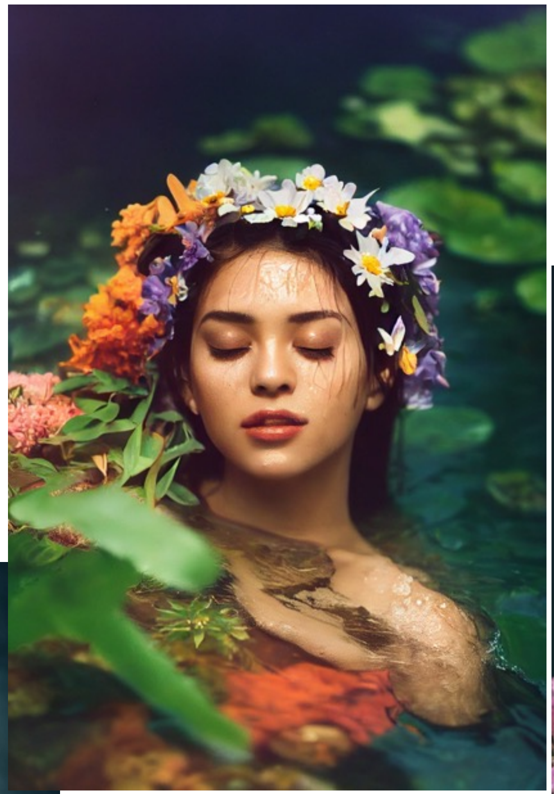
Incremental improvements increasing the realism of the generated content.

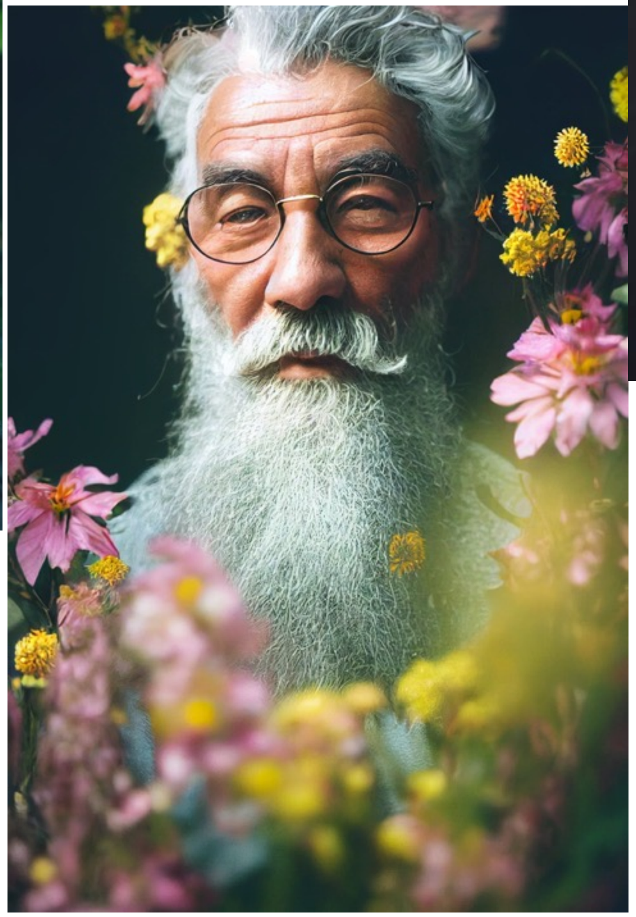
DIVERSITY AND CONTROL

Models that not only generate high fidelity but also diverse content that can be controlled by the user.

EVEN MORE DIVERSITY AND CONTROL

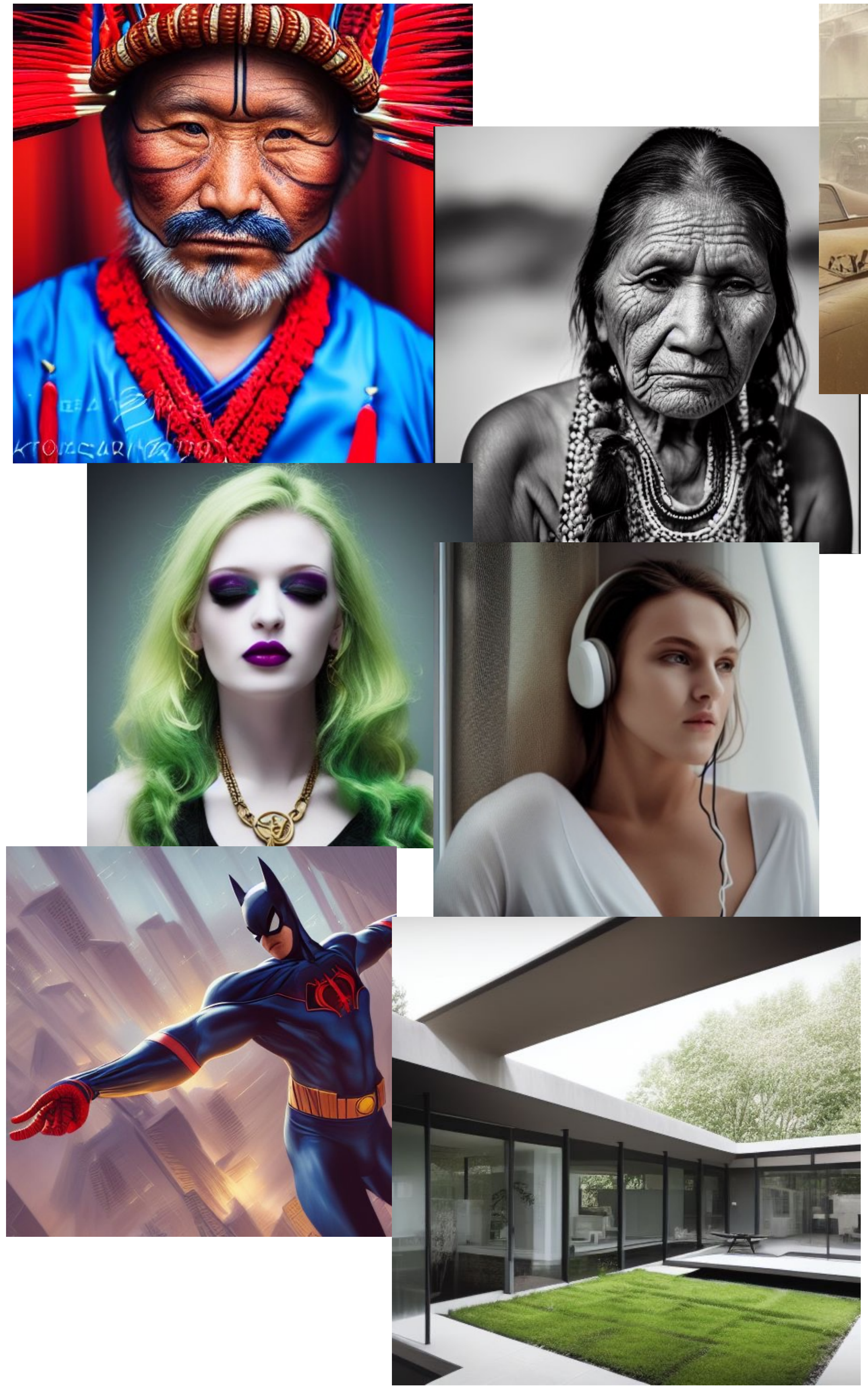
Blurring the line between digitally created art and reality

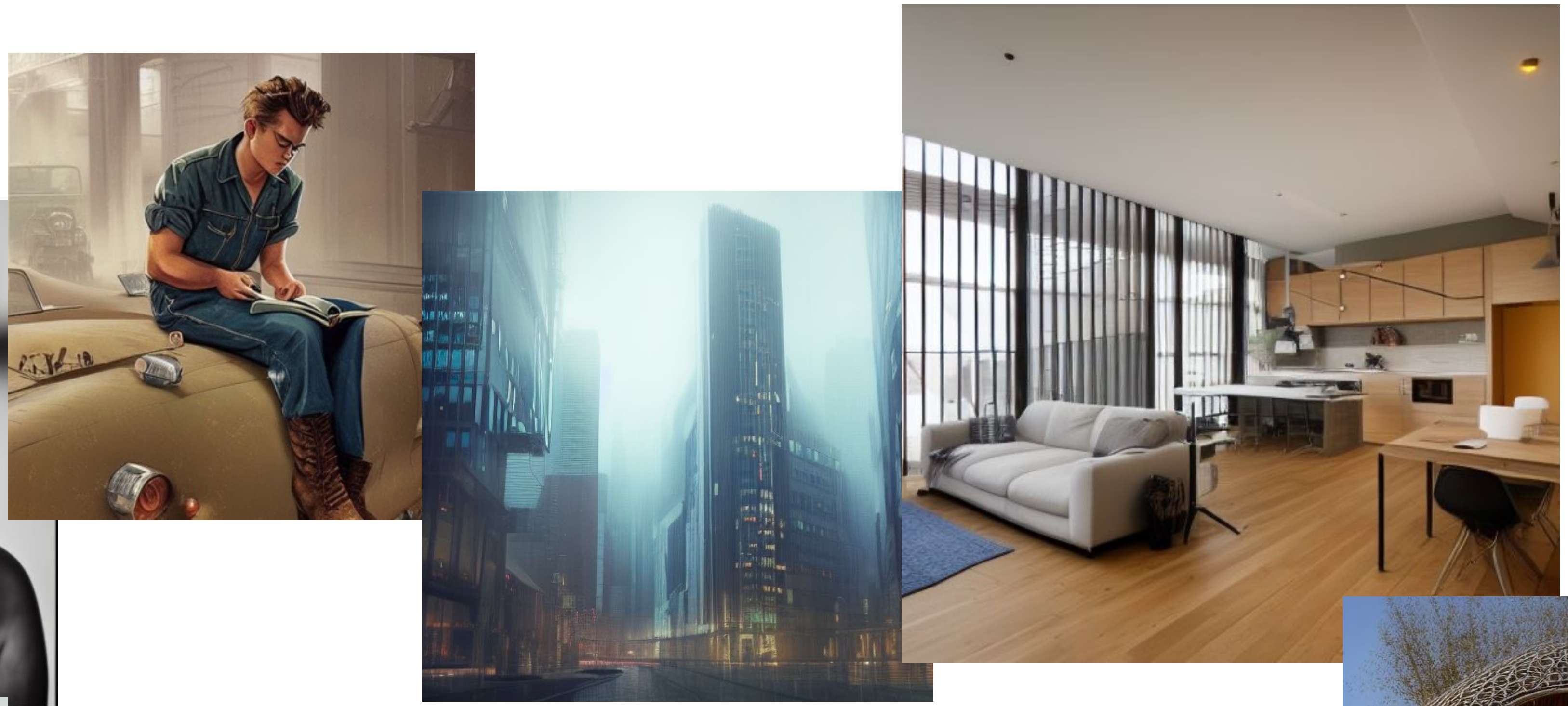


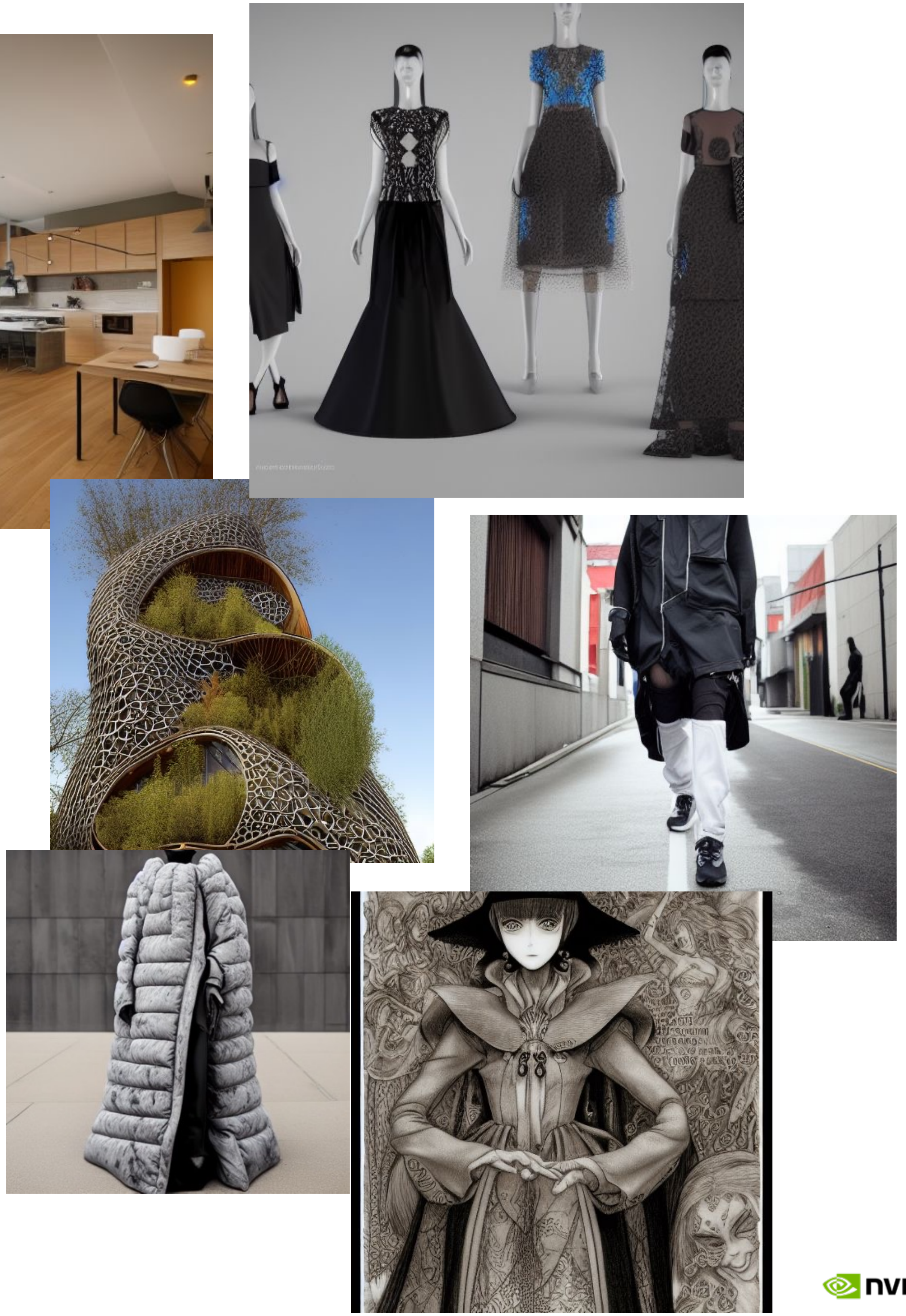


EASE OF USE

Critical mass





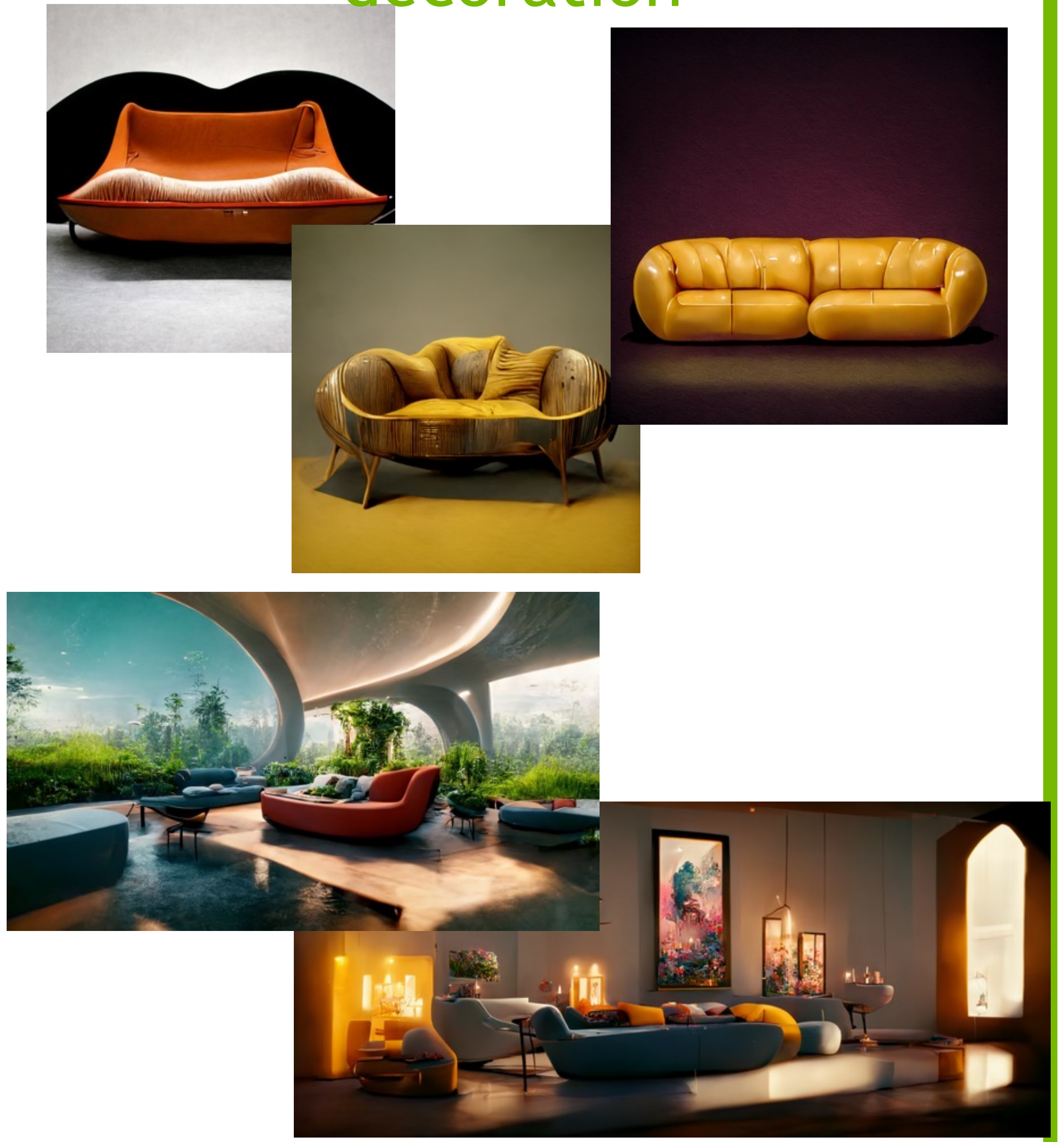


NVIDIA

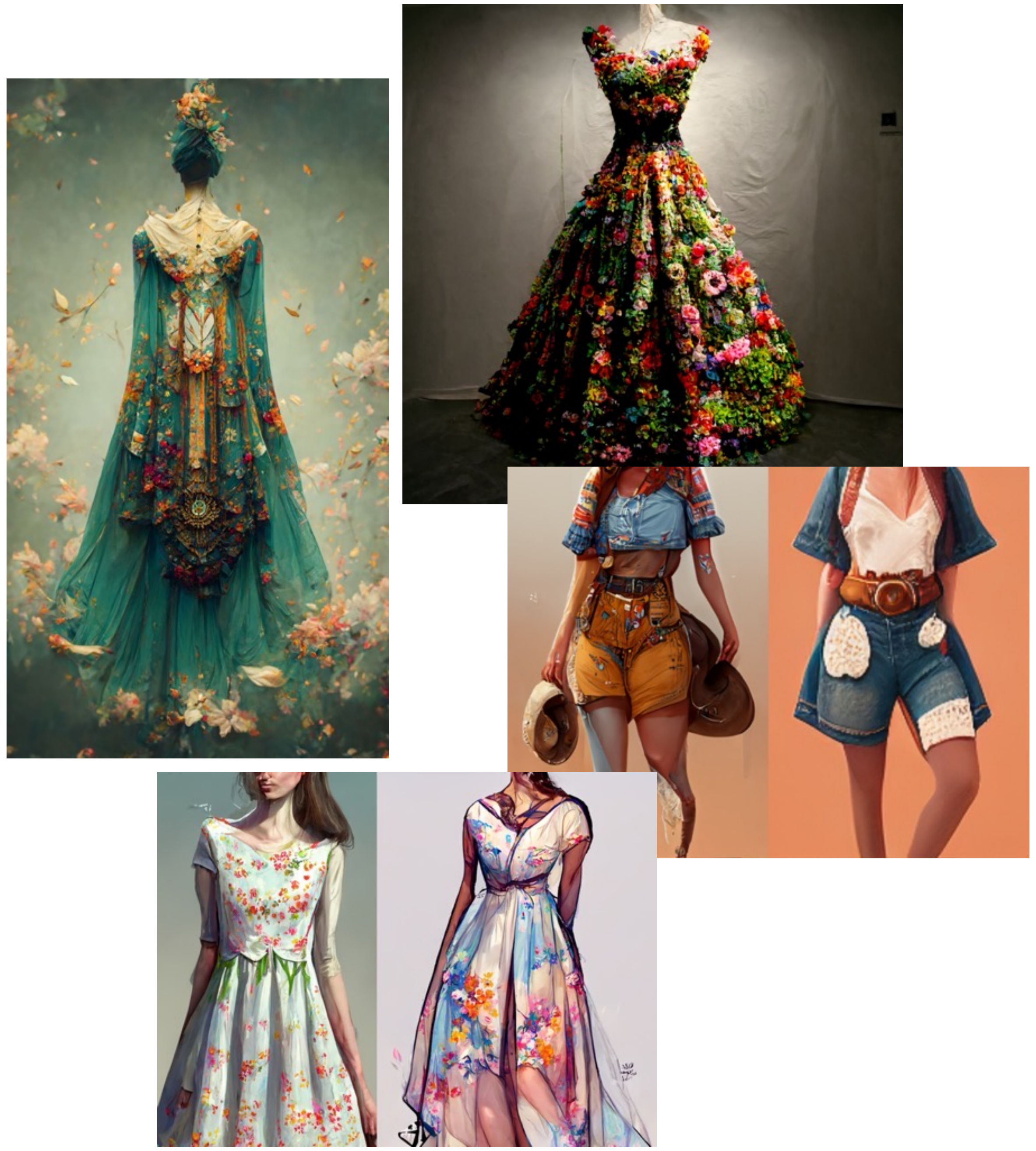
ANY FORM OF DESIGN

From Interior decoration to… Architecture

Furniture and interior decoration



Fashion



Architecture

New York Times: A.I.-Generated Art Is Already Transforming Creative Work
https://www.nytimes.com/2022/10/21/technology/ai-generated-art-jobs-dall-e-2.html

ANY FORM OF DESIGN

...to Automotive and more

Automotive

Game development

Text to image on Stable Diffusion, using the prompt: 'magical off world dreamscape'

Image-to-image translation using Stable Diffusion

Biology / Chemistry / Material Science / Scientific Visualization / ???

ROBOTICS

Planning and Imagination

Publications > StructDiffusion: Language-Guided Creation of Physically-Valid Structures using Unseen Objects

StructDiffusion: Language-Guided Creation of Physically-Valid Structures using Unseen Objects

Robots operating in human environments must be able to rearrange objects into semantically-meaningful configurations, even if these objects are previously unseen. We focus on the problem of building physically-valid structures without step-by-step instructions.

We propose StructDiffusion, which combines a diffusion model and an object-centric transformer to construct structures given partial-view point clouds and high-level language goals, such as *"set the table"* and *"make a line"*.

StructDiffusion improves success rate on assembling physically-valid structures out of unseen objects by on average 16% over an existing multi-modal transformer model, while allowin us to use one multi-task model to produce a wider range of different structures. We show experiments on held-out objects in both simulation and on real-world rearrangement tasks.

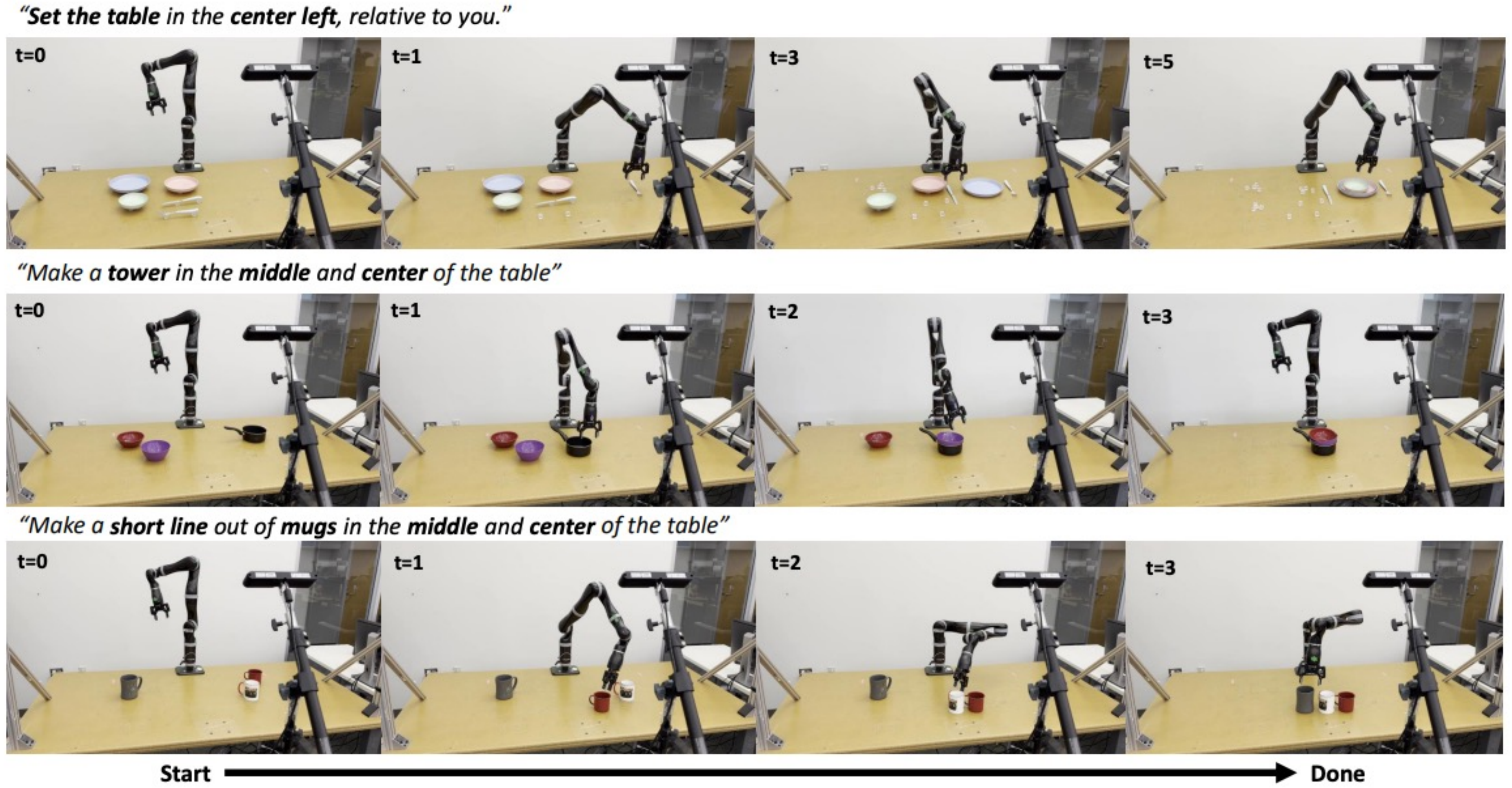


Fig. 1: Real-world rearrangement with unseen objects, given a language instruction. We use StructDiffusion to predict possible goals that satisfy physical constraints such as avoiding collisions between objects. At the core of StructDiffusion is an object-centric multimodal transformer backbone combined with a diffusion model, capable of sampling diverse high-level motion goals for language-guided rearrangement.

ProgPrompt: Generating Situated Robot Task Plans using Large Language Models

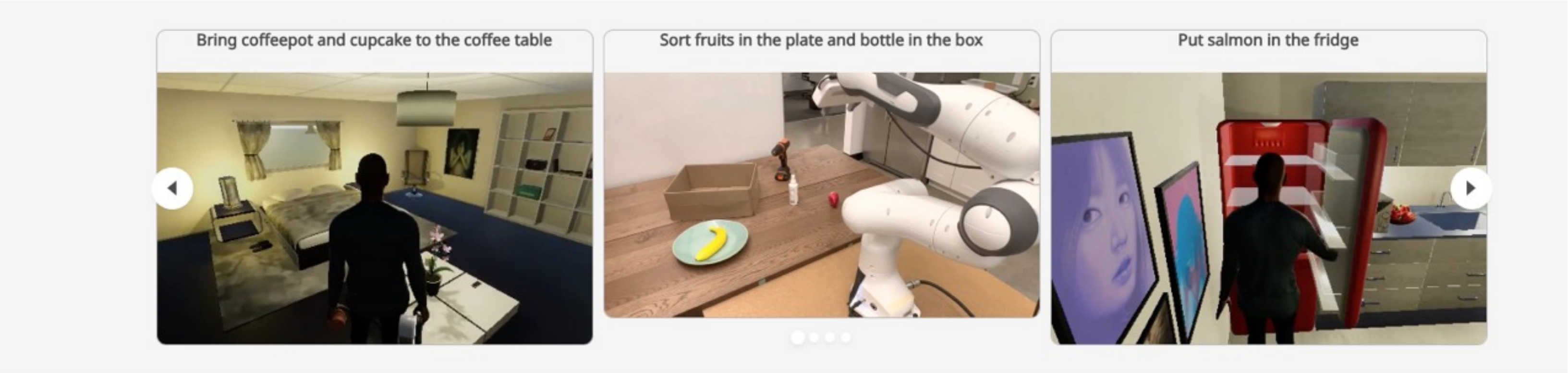
ICRA 2023

Extended version in Autonomous Robots 2023

Ishika Singh[1], Valts Blukis[2], Arsalan Mousavian[2], Ankit Goyal[2], Danfei Xu[2], Jonathan Tremblay[2], Dieter Fox[2], Jesse Thomason[1], Animesh Garg[2]

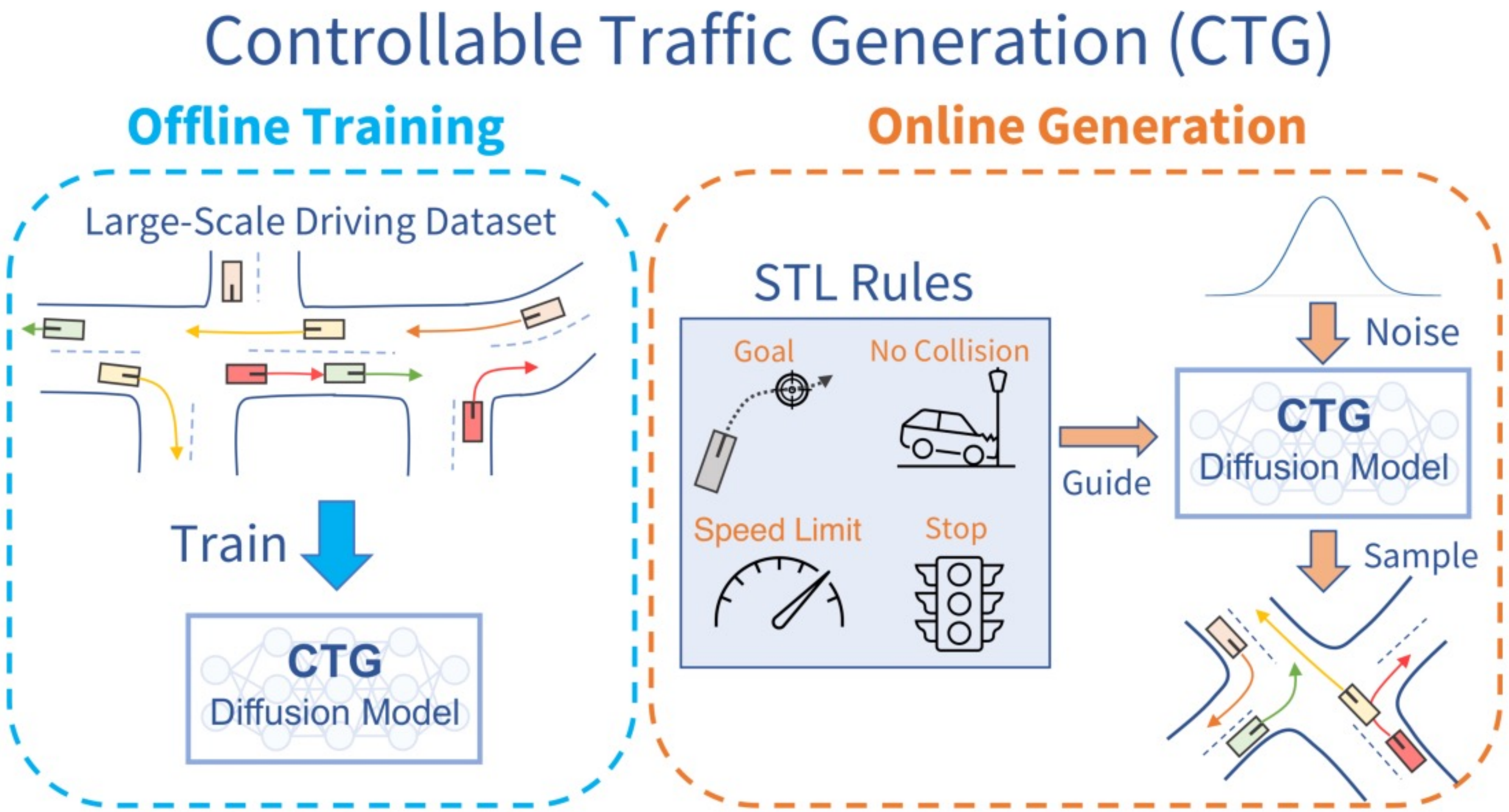
[1]University of Southern California, [2]NVIDIA

arXiv | Video | Code | Poster | ? FAQs



SIMULATION

Guided Conditional Diffusion for Controllable Traffic Simulation



Controllable and realistic traffic simulation is critical for developing and verifying autonomous vehicles. Typical heuristic-based traffic models offer flexible control to make vehicles follow specific trajectories and traffic rules. On the other hand, data-driven approaches generate realistic and human-like behaviors, improving transfer from simulated to real-world traffic. However, to the best of our knowledge, no traffic model offers both controllability and realism. In this work, we develop a conditional diffusion model for controllable traffic generation (CTG) that allows users to control desired properties of trajectories at test time (e.g., reach a goal or follow a speed limit) while maintaining realism and physical feasibility through enforced dynamics. The key technical idea is to leverage recent advances from diffusion modeling and differentiable logic to guide generated trajectories to meet rules defined using signal temporal logic (STL). We further extend guidance to multi-agent settings and enable interaction-based rules like collision avoidance. CTG is extensively evaluated on the nuScenes dataset for diverse and composite rules, demonstrating improvement over strong baselines in terms of the controllability-realism tradeoff.

PHYSICS

A Physics-informed Diffusion Model for High-fidelity Flow Field Reconstruction

Dule Shu,[†,§] Zijie Li,[†,§] and Amir Barati Farimani[*,†,‡,¶]

†*Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh PA, USA*

‡*Machine Learning Department, Carnegie Mellon University, Pittsburgh PA, USA*

¶*Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh PA, USA*

§*Contributed equally to this work*

E-mail: barati@cmu.edu

Abstract

Machine learning models are gaining increasing popularity in the domain of fluid dynamics for their potential to accelerate the production of high-fidelity computational fluid dynamics data. However, many recently proposed machine learning models for high-fidelity data reconstruction require low-fidelity data for model training. Such requirement restrains the application performance of these models, since their data reconstruction accuracy would drop significantly if the low-fidelity input data used in model test has a large deviation from the training data. To overcome this restraint, we propose a diffusion model which only uses high-fidelity data at training. With different configurations, our model is able to reconstruct high-fidelity data from either a regular low-fidelity sample or a sparsely measured sample, and is also able to gain an accuracy increase by using physics-informed conditioning information from a known partial differential equation when that is available. Experimental results demonstrate that our model can produce accurate reconstruction results for 2d turbulent flows based on different input sources without retraining.

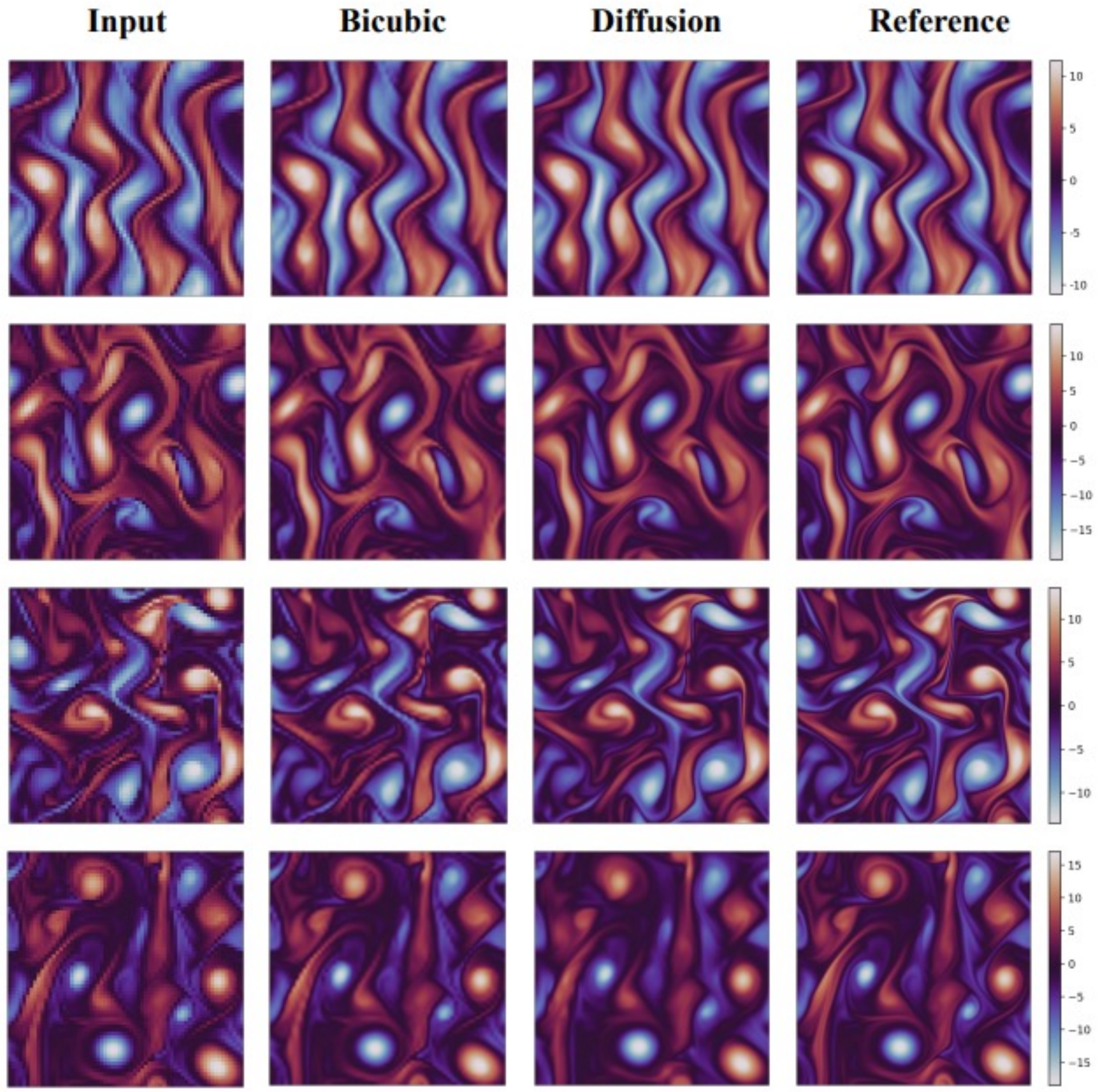
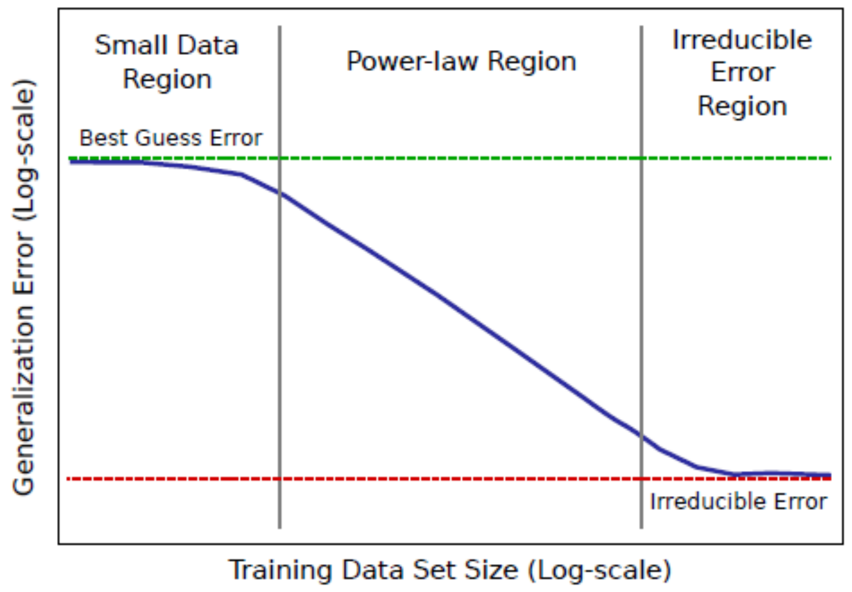


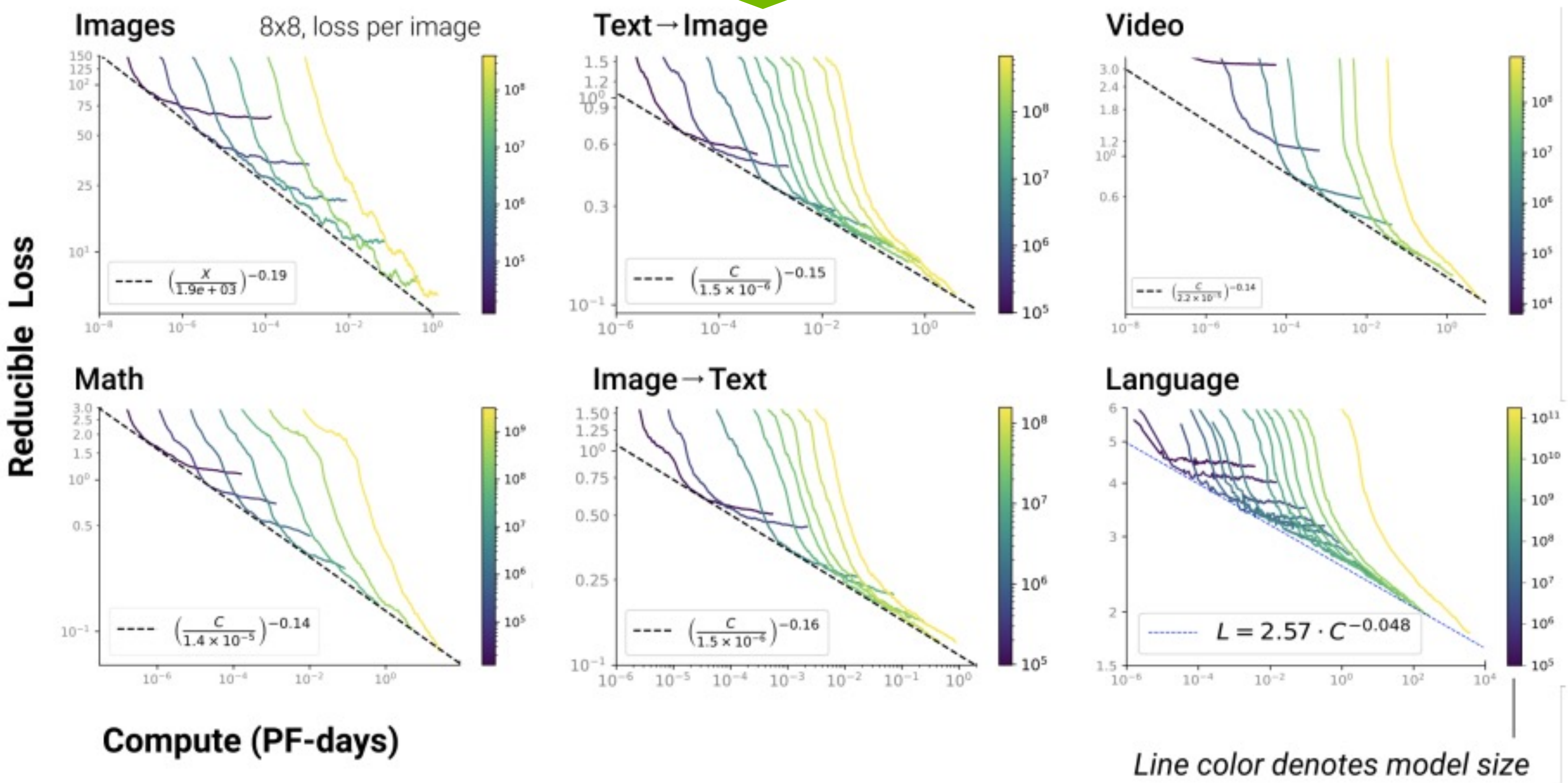
Figure 3: Qualitative comparison of different upsampling methods on 4x upsampling task.

Other Modalities

EMPIRICAL EVIDENCE

The Scaling Laws for Generative models



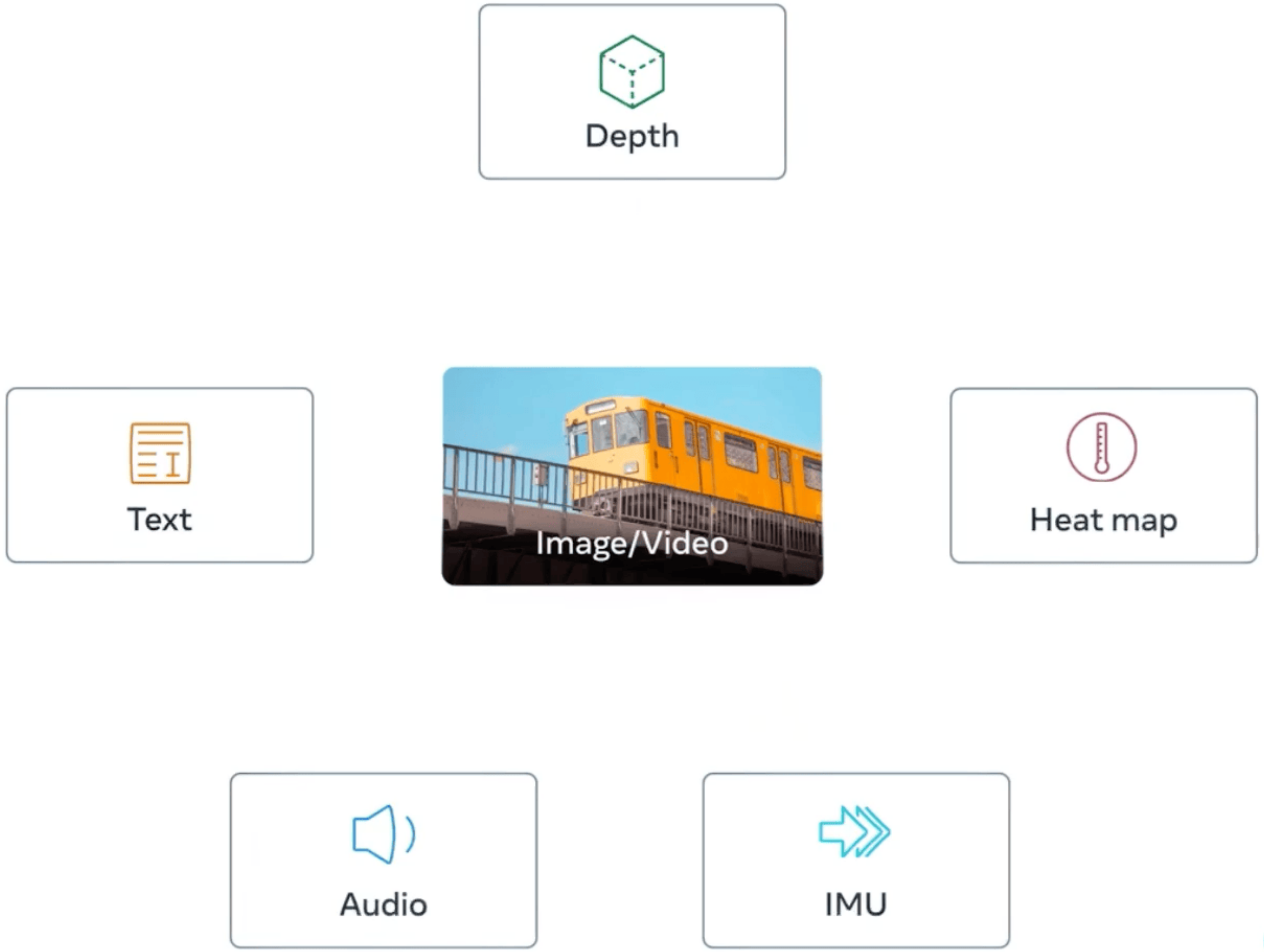


Henighan, Tom, et al. **Scaling laws for autoregressive generative modeling.** arXiv preprint arXiv:2010.14701 (2020).

Multimodal architectures

This is just the first wave

Rise of multimodal architectures



Meta AI

Simplicity of multimodal architectures

LLAVA example

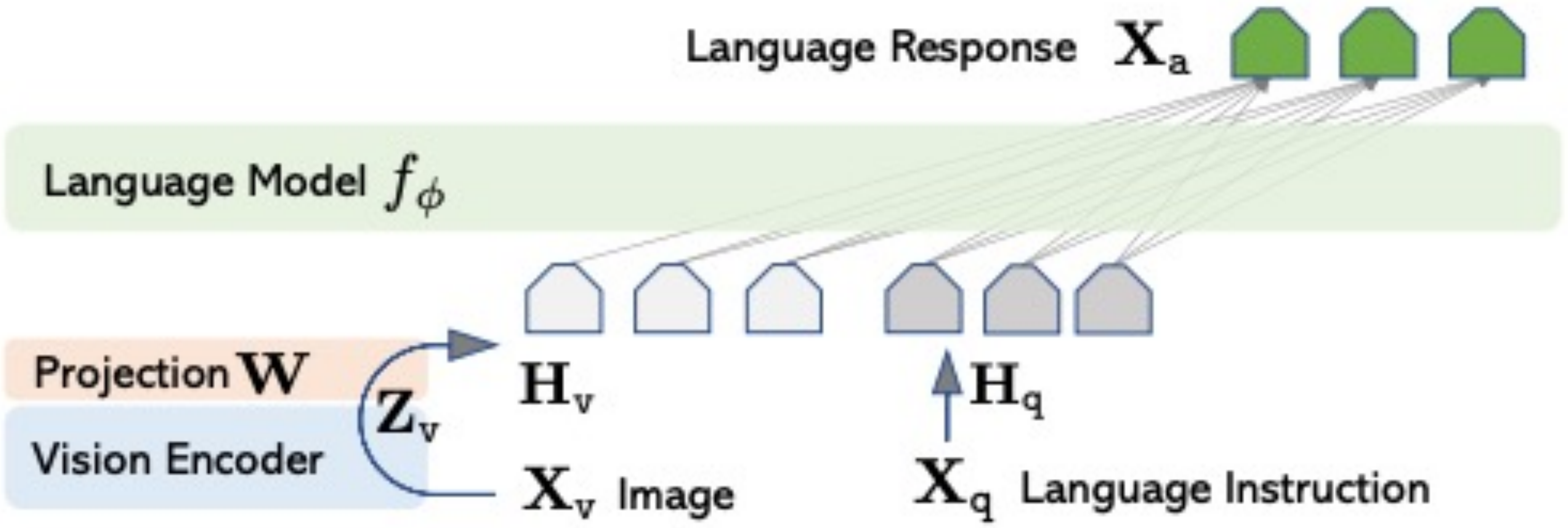


Figure 1: LLaVA network architecture.

This is just the first wave

Rise of multimodal architectures

Large Language Models are Large

LLAMA 2 TRAINING TIME

Hypothetical Training Time on single NVIDIA A100 GPUs

Single GPU

meta-llama

24 years +

NVIDIA

LLAMA 2 TRAINING TIME

Training Time on NVIDIA A100 GPUs

DiRAC: Tursa

meta-llama

157 days

GOING BIGGER

The challenge

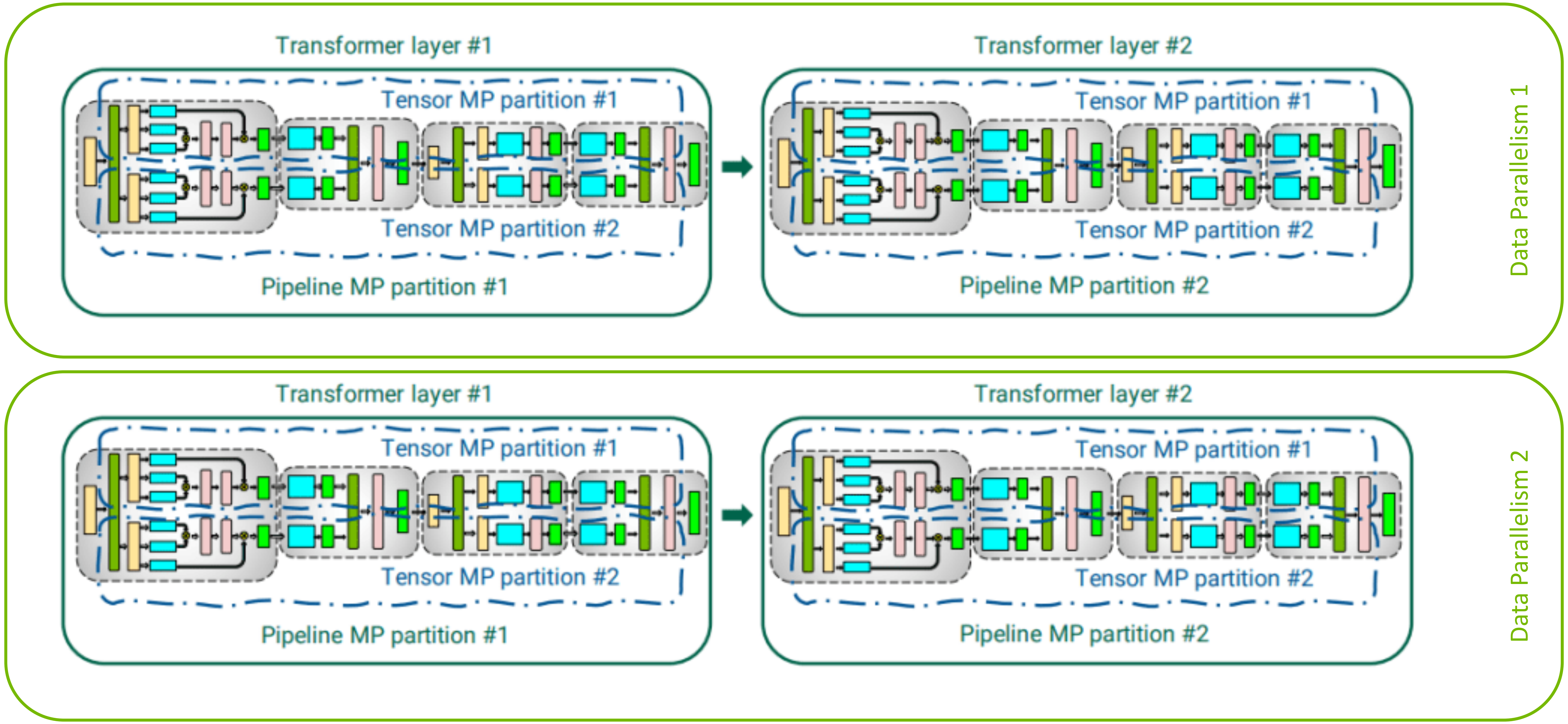
Consider 1 billion parameters model in **FP16** and do the math:

- Data representation: Weights and Gradients in FP16
- Adam optimizer: Store 12 bytes per weight in FP16

$$\underbrace{10^9}_{\text{1 billion parameters}} * (\underbrace{2B}_{\text{2 bytes per weight}} + \underbrace{2B}_{\text{2 bytes per gradient}} + \underbrace{12B}_{\text{12 bytes per optimizer state}}) = \mathbf{14.90GB}$$

DEALING WITH MEMORY CONSTRAINTS

Various Forms of Parallelism



https://github.com/NVIDIA/Megatron-LM/

Whole platform approach

MODULUS
MONAI
RIVA
MAXINE
NEMO
MERLIN
CUOPT
MORPHEUS
TOKKIO
AVATAR
DRIVE
ISAAC
METROPOLIS
HOLOSCAN
AI APPLICATION FRAMEWORK
PLATFORMS
NVIDIA HPC
NVIDIA AI
NVIDIA Omniverse
ACCELERATION LIBRARIES
cuNumeric
CV-CUDA
cuQuantum
Parabricks
Sionna
JetPack
RAPIDS
Spark
cuDNN
cuGraph
TensorRT
Triton
DeepStream
Flare
DOCA
Mag IO
Aerial
CLOUD-TO-EDGE DATACENTER-TO-ROBOTIC SYSTEMS
RTX
DGX
HGX
EGX
OVX
Super POD
AGX
3 CHIPS
GPU
CPU
DPU

Today focusing on infrastructure

Lessons from the NVIDIA AI Journey

Industry-leading expertise gained from our most important endeavors

2016
- DGX SATURNV Initial Install
- 125 Node Cluster TOP500 #28
- DGX-1 Launch

2018
- Expansion to 1,468 Nodes
- "Circe"36 Node DGX SuperPOD
- MLPerf
- DGX-2 Launch

2019
- TOP500 #20
- Expansion to 1,800 Nodes
- 96 Node DGX SuperPOD
- TOP500 #22

2020
- Green500 #1: "NVIDIA DGX SuperPOD"
- TOP500 #5: "Selene DGX SuperPOD"
- DGX A100 Launch
- Expansion to 2,000 Nodes
- "Selene" 560 Node DGX A100 SuperPOD

Jun 2021
- Green500 #5: "NVIDIA DGX SuperPOD"
- TOP500 #6: "Selene DGX SuperPOD"
- 14 systems in the TOP500 and Green500 built on NVIDIA DGX BasePOD and DGX SuperPOD

Nov 2021
- Green500 #3: "NVIDIA DGX SuperPOD"
- TOP500 #6: "Selene DGX SuperPOD"
- 17 systems in the TOP500 and Green500 built on NVIDIA DGX BasePOD and DGX SuperPOD

Mar 2022
- DGX H100 Launch

Nov 2022
- Green500 #23: "NVIDIA DGX SuperPOD
- TOP500 #9: "Selene DGX SuperPOD"
- 17 systems in the TOP500 and Green500 built on NVIDIA DGX BasePOD and DGX SuperPOD

Mar 2023
- DGX Cloud Launch

May 2023
- DGX GH200 Launch

NGC Research & Development Autonomous Cars Robotics RTX Graphics

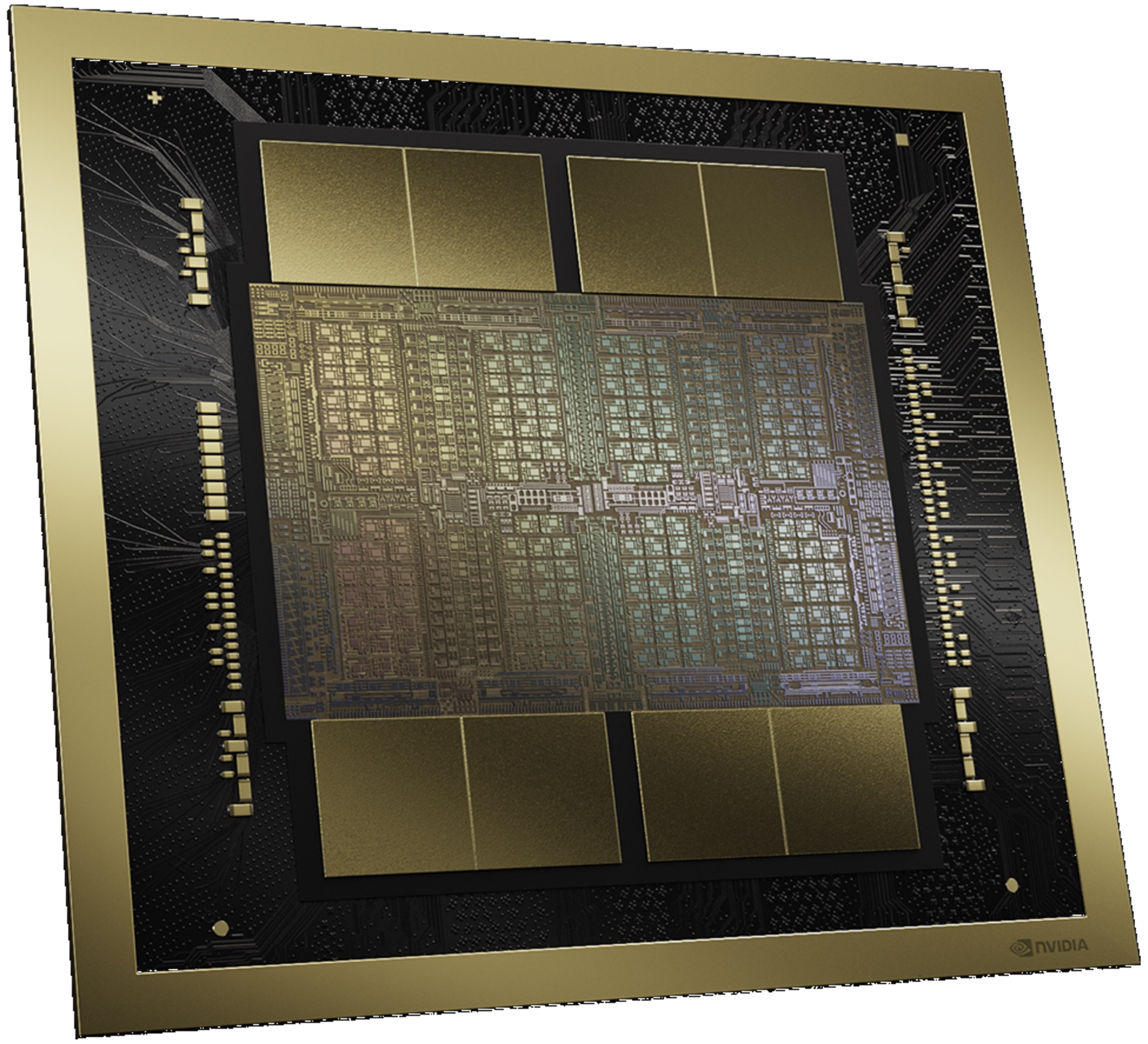
- Designing for predictable performance at scale
- Operations/Infrastructure manageability & support
- AI workflow management / data science productivity

Going across the stack

Of course the GPU

Announcing NVIDIA Blackwell

The Engine of the New Industrial Revolution



Built to Democratize Trillion-Parameter AI

20 PetaFLOPS of AI performance on a single GPU

4X Training | 30X Inference | 25X Energy Efficiency & TCO

Expanding AI Datacenter Scale to beyond 100K GPUs

AI SUPERCHIP
208B Transistors

2nd GEN TRANSFORMER ENGINE
FP4/FP6 Tensor Core

5th GENERATION NVLINK
Scales to 576 GPUs

RAS ENGINE
100% In-System
Self-Test

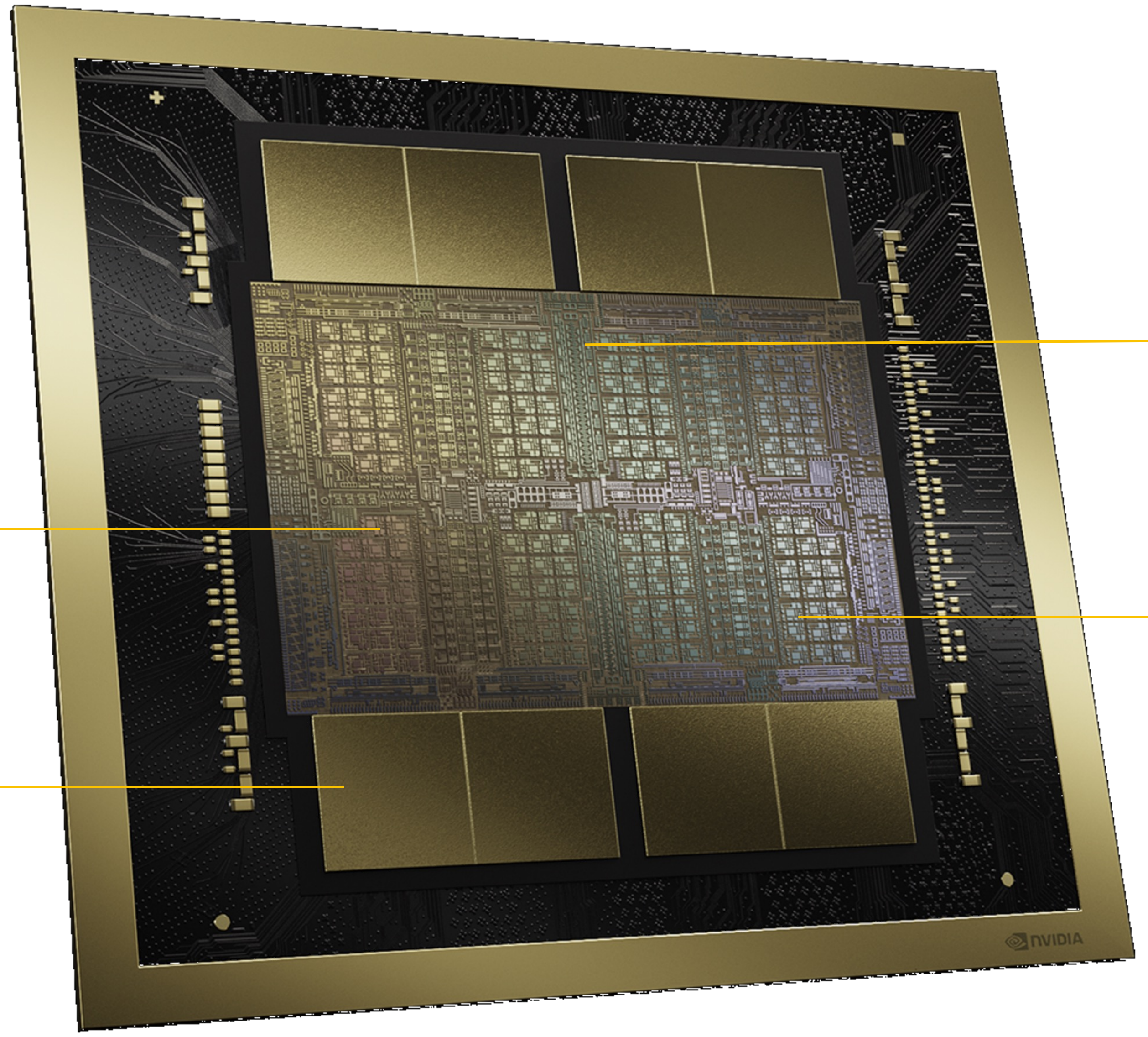
SECURE AI
Full Performance
Encryption & TEE

DECOMPRESSION ENGINE
800 GB/s

NVIDIA

New Class of AI Superchip

The Two Largest Dies Possible—Unified as One GPU



Reticle-sized Die 1

Fast Memory
192GB
HBM3e

2 reticle-limited dies operate as One Unified CUDA GPU

NV-HBI 10TB/s High Bandwidth Interface

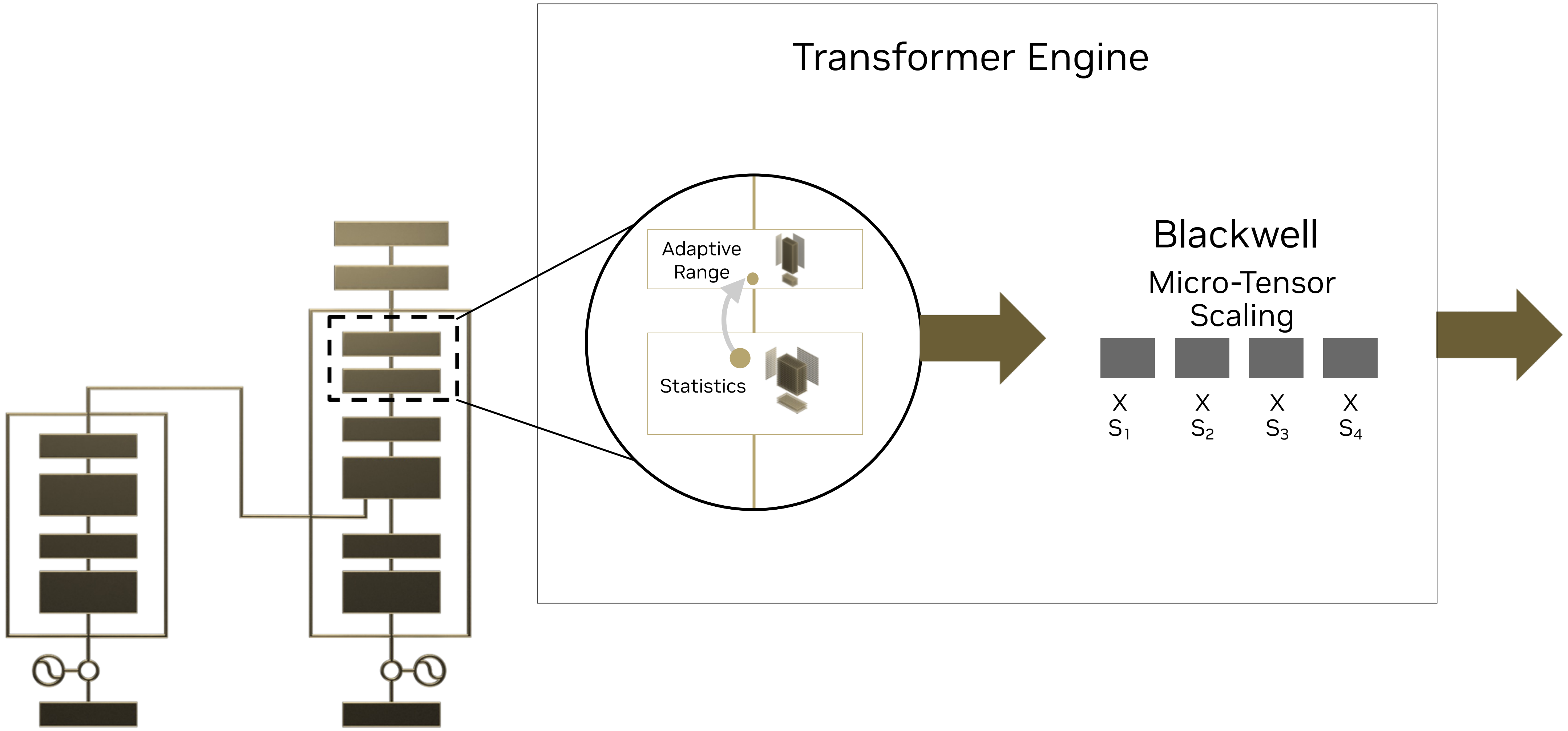
Full performance. No compromises

Reticle-sized Die 2

10 PetaFLOPS FP8 | 20 PetaFLOPS FP4
192GB HBM3e | 8 TB/sec HBM Bandwidth | 1.8TB/s NVLink

2nd Generation Transformer Engine

Accelerating Throughput with Intelligent 4-Bit Precision



Enabling FP4 AI Inference

2x Compute

2x Bandwidth

2x Model Size

LLAMA 2 training time

Hypothetical Training Time on single NVIDIA A100 GPUs

Single GPU

meta-llama

24 years +

Adapting to even larger neural networks

NVIDIA Grace CPU

Building Block of the Superchip

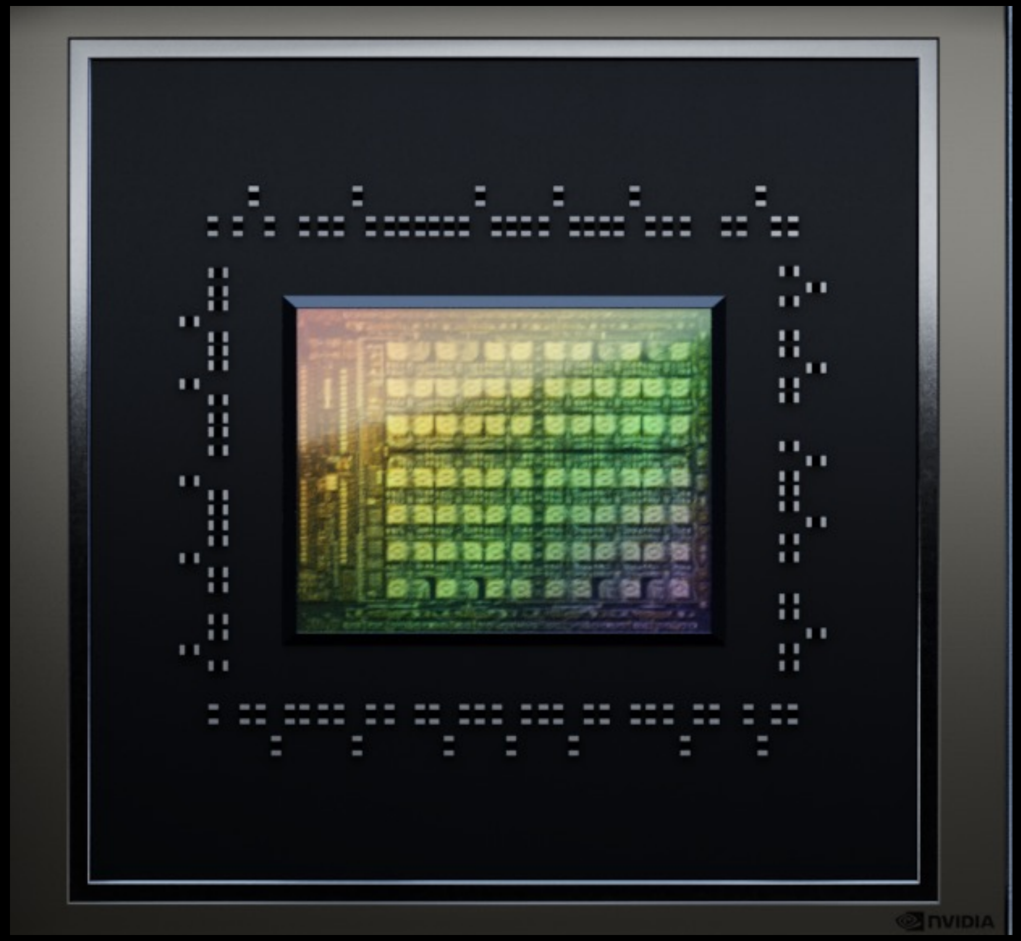
High Performance Power Efficient Cores
72 flagship Arm Neoverse V2 Cores with
SVE2 4x128b SIMD per core

Fast On-Chip Fabric
3.2 TB/s of bisection bandwidth connects
CPU cores, NVLink-C2C, memory, and system IO

High-Bandwidth Low-Power Memory
Up to 480 GB of data center enhanced LPDDR5X Memory that
delivers up to 500 GB/s of memory bandwidth

Coherent Chip-to-Chip Connections
NVLink-C2C with 900 GB/s bandwidth for coherent
connection to CPU or GPU

Industry Leading Performance Per Watt
Up to 2X perf / W over today's leading servers

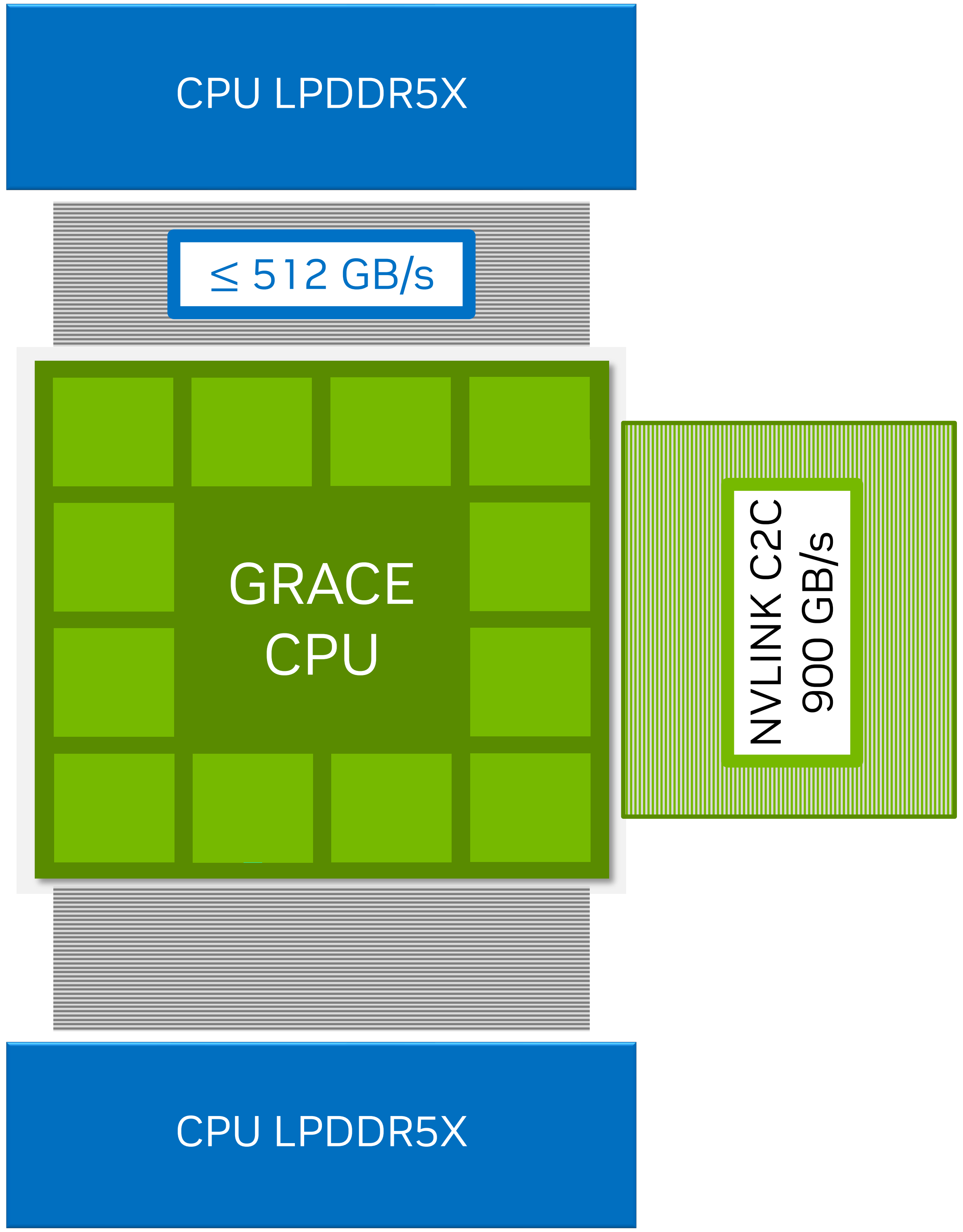


NVIDIA Grace CPU

NVLINK-C2C

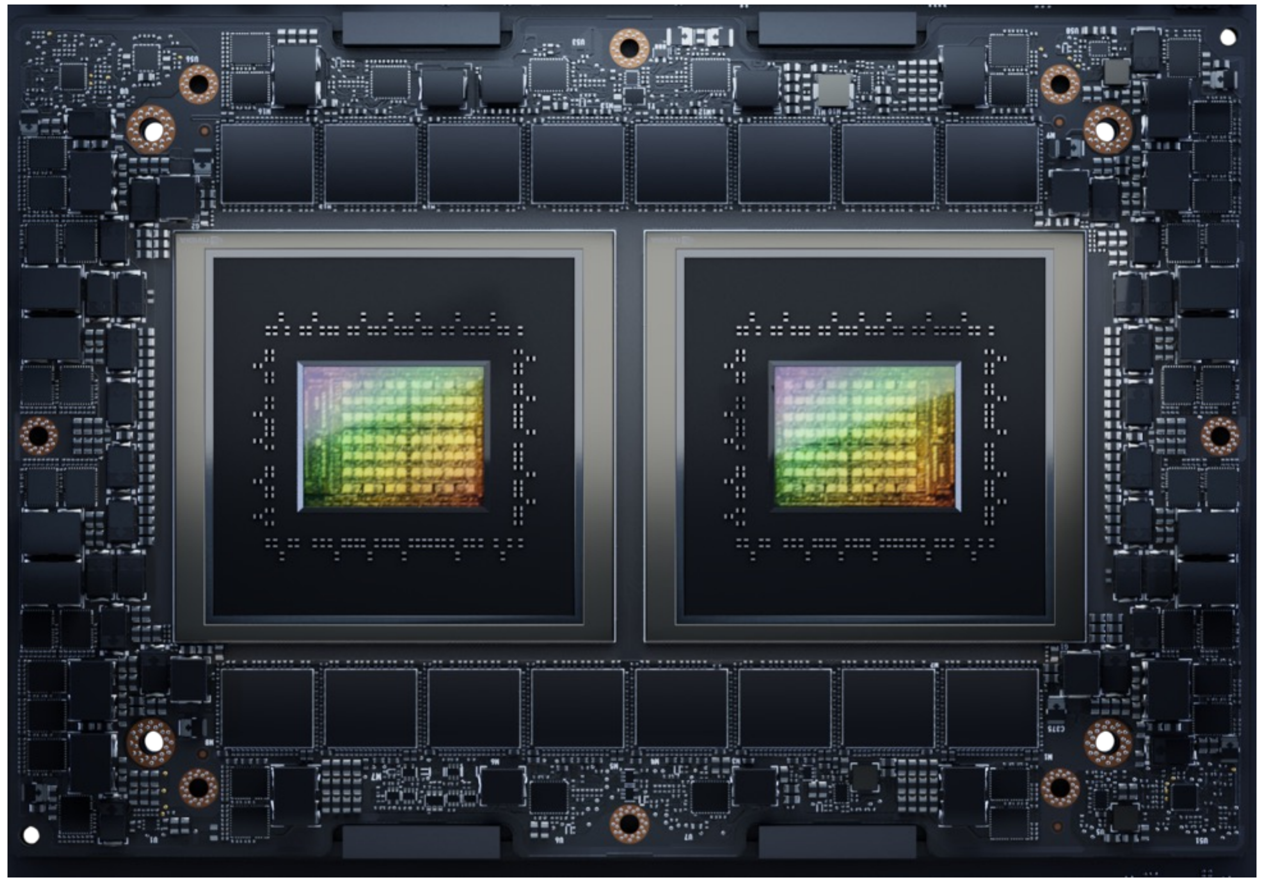
High Speed Chip to Chip Interconnect

- Creates Grace Hopper and Grace Superchips
- Removes the typical cross-socket bottlenecks
- Up to 900GB/s of raw bidirectional BW
  - Same BW as GPU to GPU NVLINK on Hopper
- Low power interface - 1.3 pJ/bit
  - More than 5x more power efficient than PCIe
- Enables coherency for both Grace and Grace Hopper superchips



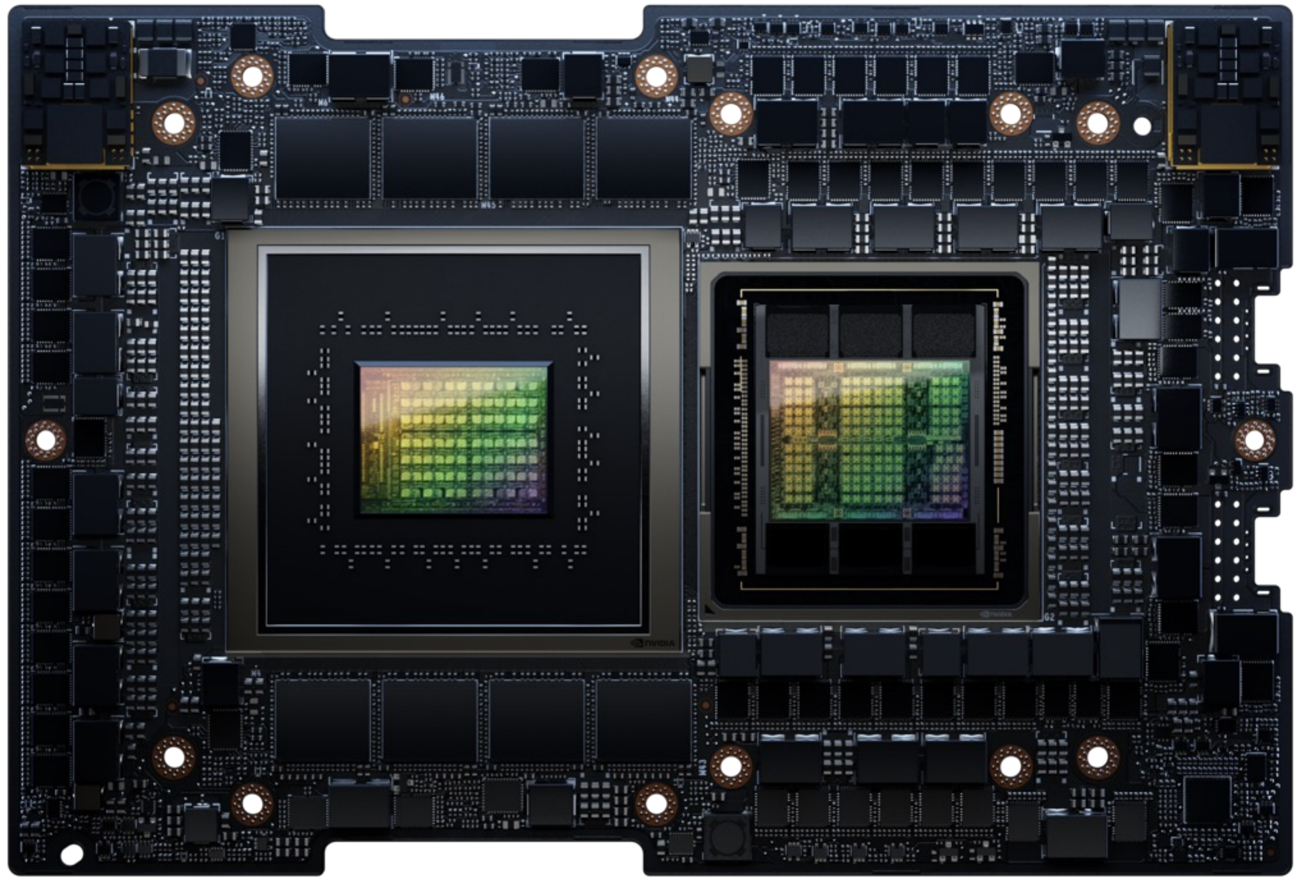
NVIDIA Grace for Cloud, AI and HPC Infrastructure

Grace CPU Superchip
CPU Computing



CPU-based applications where absolute performance, energy efficiency, and data center density matter, such as scientific computing, data analytics, enterprise and hyperscale computing applications

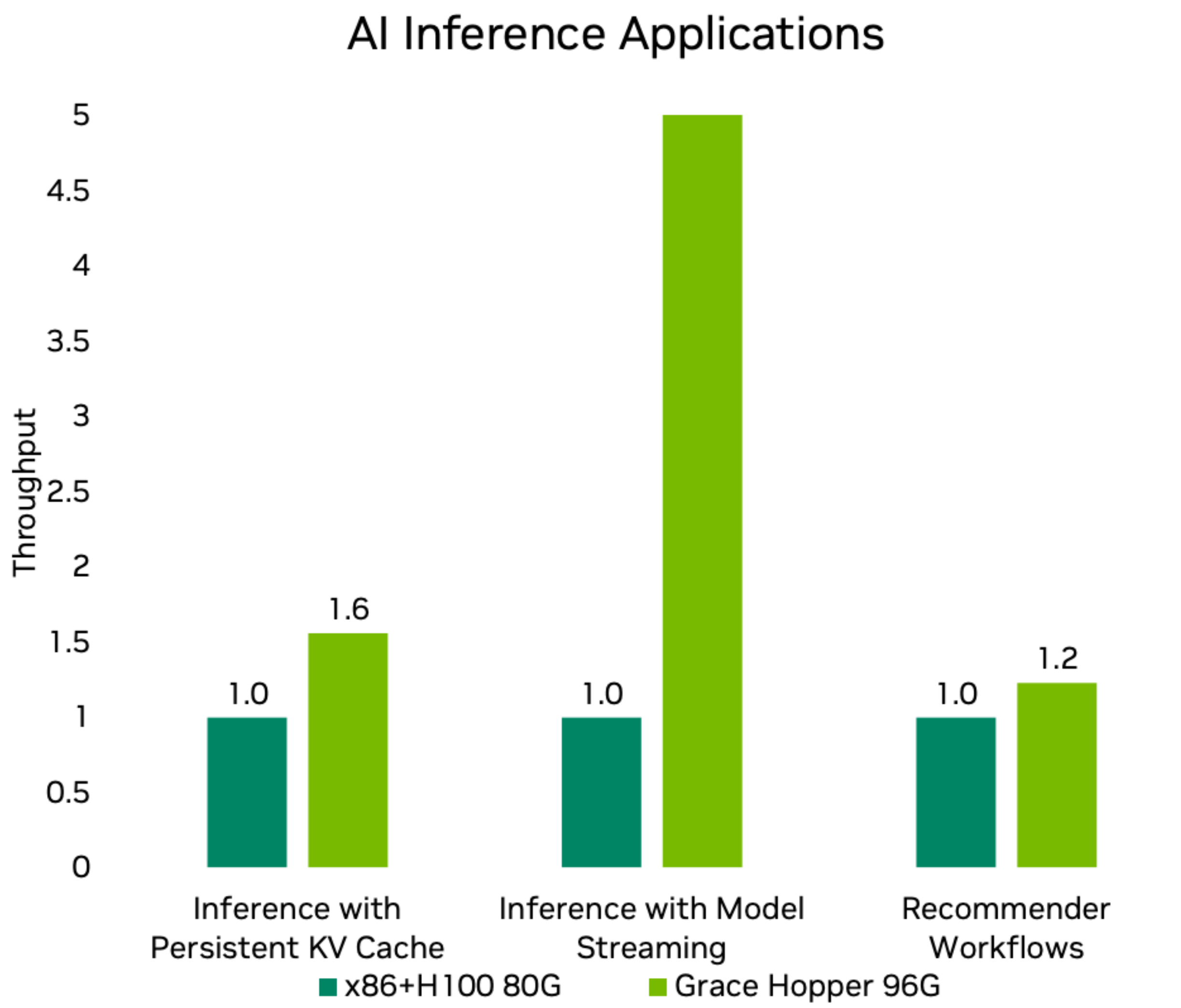
GH200 Grace Hopper Superchip
Large Scale AI & HPC

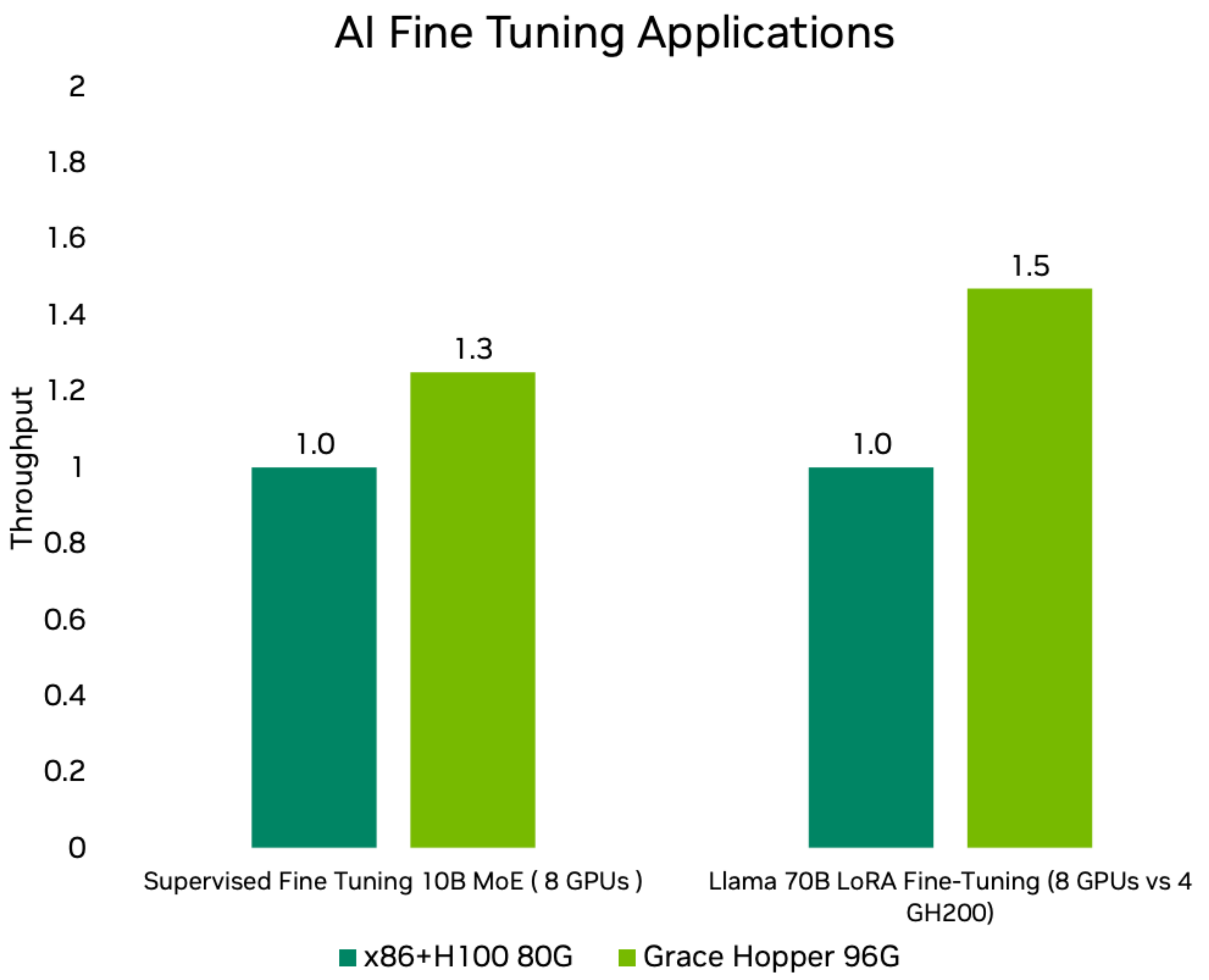


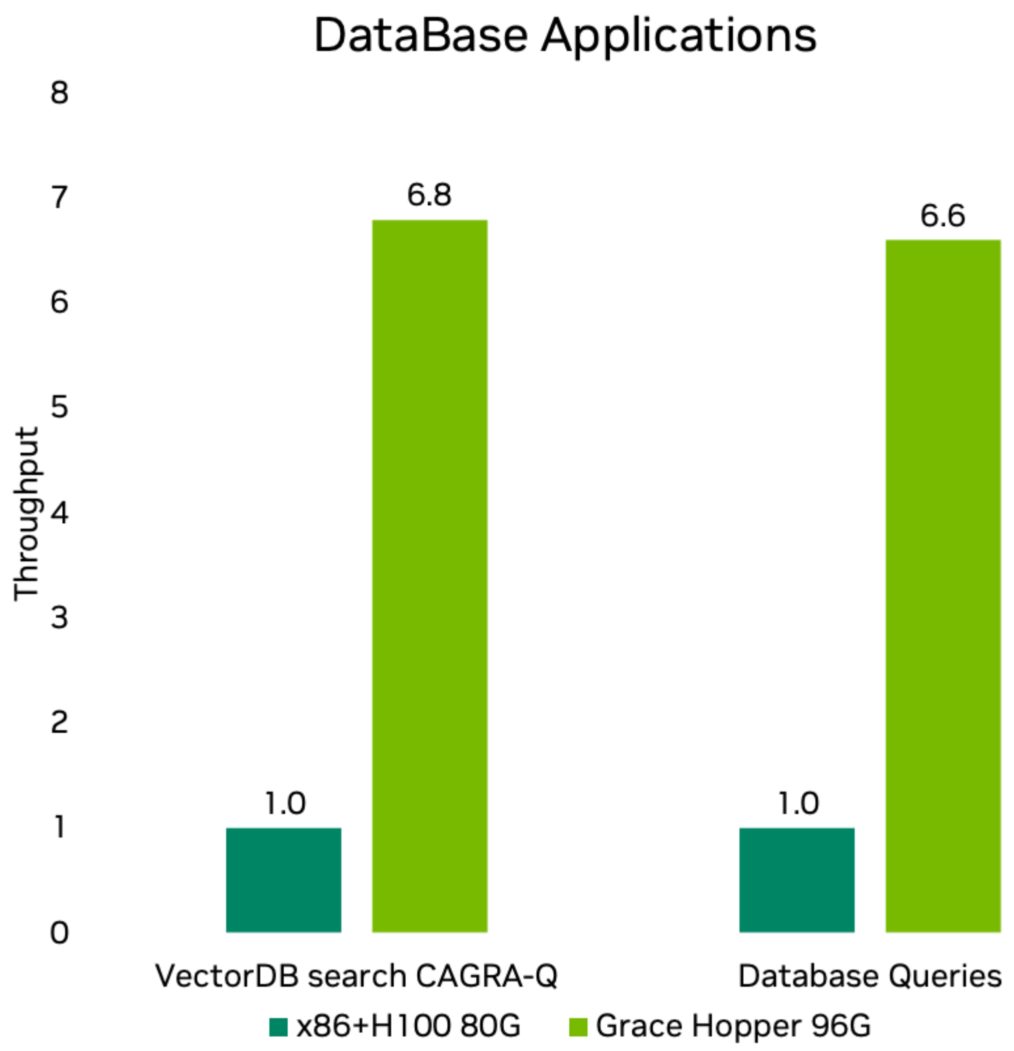
Accelerated applications where CPU performance and system memory size and bandwidth are critical; tightly coupled CPU & GPU for flagship AI & HPC. Most versatile compute platform for scale out.

Grace Hopper Performance sneak peek

Improved GPU utilization for AI applications







GH200: Grace CPU 480GB LPDDR5, H100 96GB HBM3
DGX H100 80GB HBM3

LLAMA 2 training time

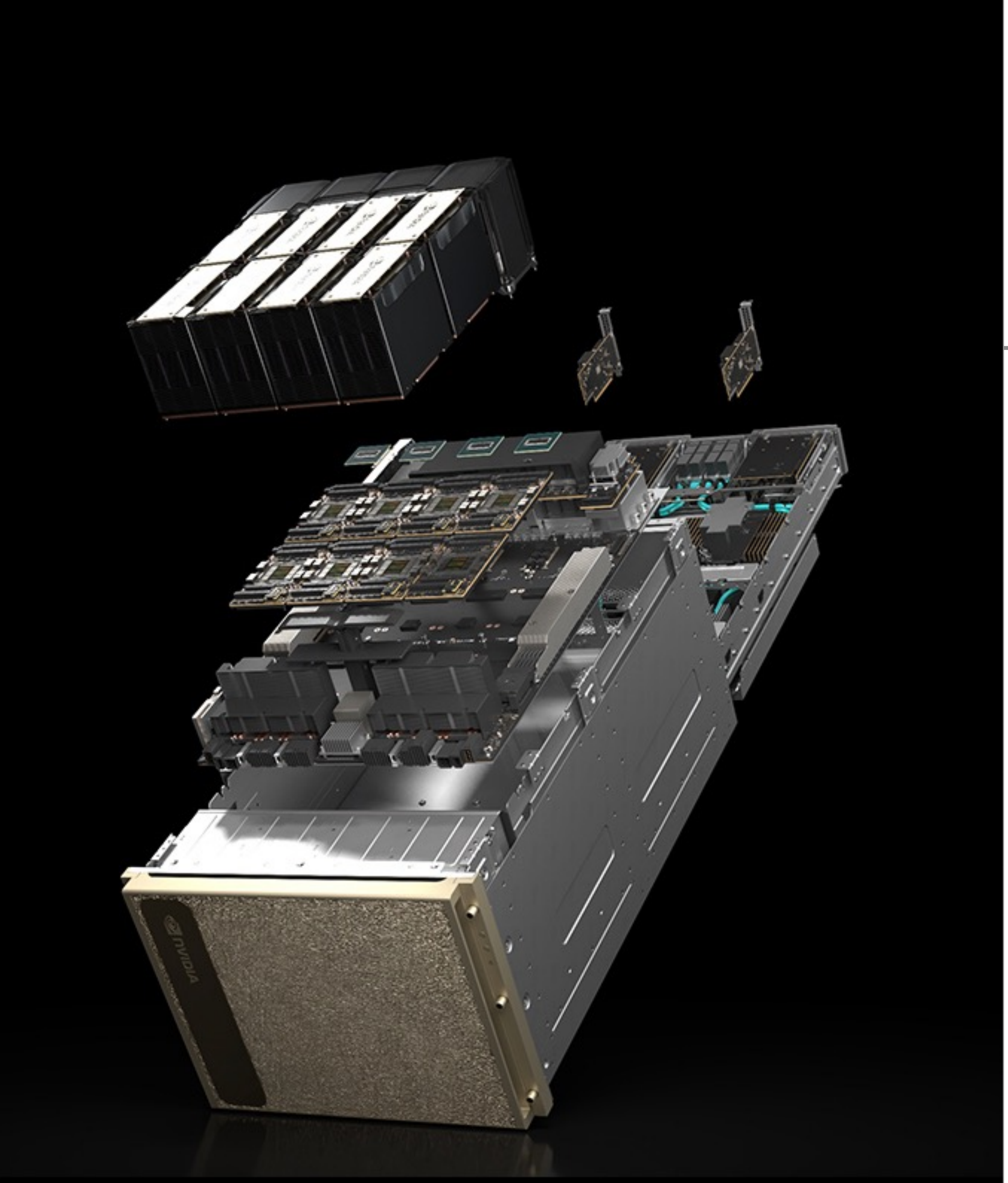
Hypothetical Training Time on single NVIDIA A100 GPUs

Single GPU

meta-llama

24 years +

Beyond a single GPU



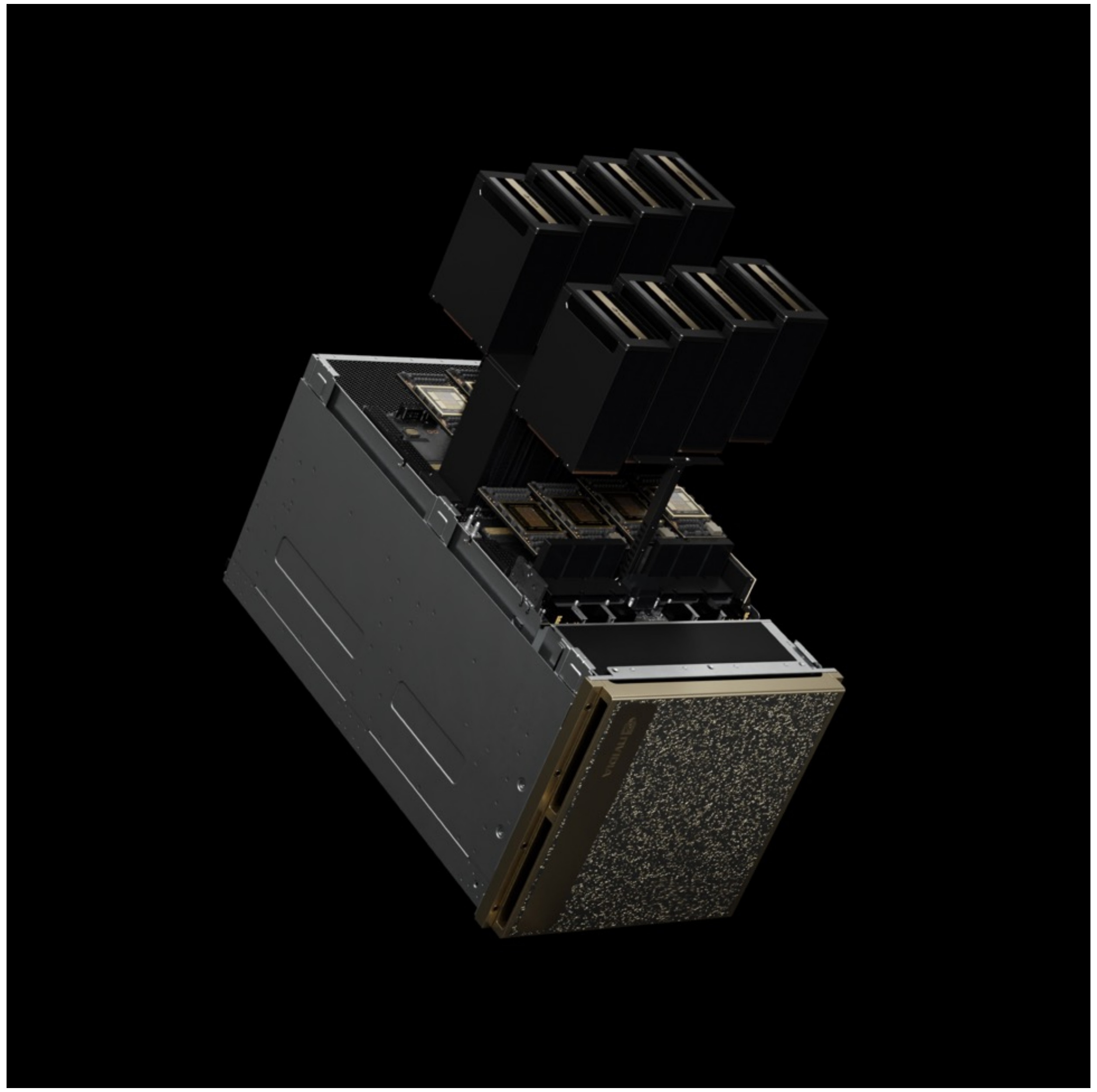
NVIDIA DGX H100: The Proven Choice for Enterprise AI

The gold standard for AI infrastructure

- 8x NVIDIA H100 GPUs With 640 Gigabytes of Total GPU Memory
  - 18x NVIDIA NVLink connections per GPU, 900 gigabytes per second of bidirectional GPU-to-GPU bandwidth
  - 24 TB/s memory bandwidth
- 4x NVIDIA NVSwitches
  - 7.2 terabytes per second of bidirectional GPU-to-GPU bandwidth, 1.5X more than previous generation
- 10x NVIDIA ConnectX-7 400 Gigabits-Per-Second Network Interface
  - 1 terabyte per second of peak bidirectional network bandwidth
- Dual 56-core 4th Gen Intel® Xeon® Scalable Processors and 2 TB System Memory
  - Powerful CPUs and massive system memory for the most intensive AI jobs
- 30 Terabytes NVMe SSD
  - High speed storage for maximum performance
- 32 petaFLOPS AI performance

DGX B200

The foundation of the modern AI data center

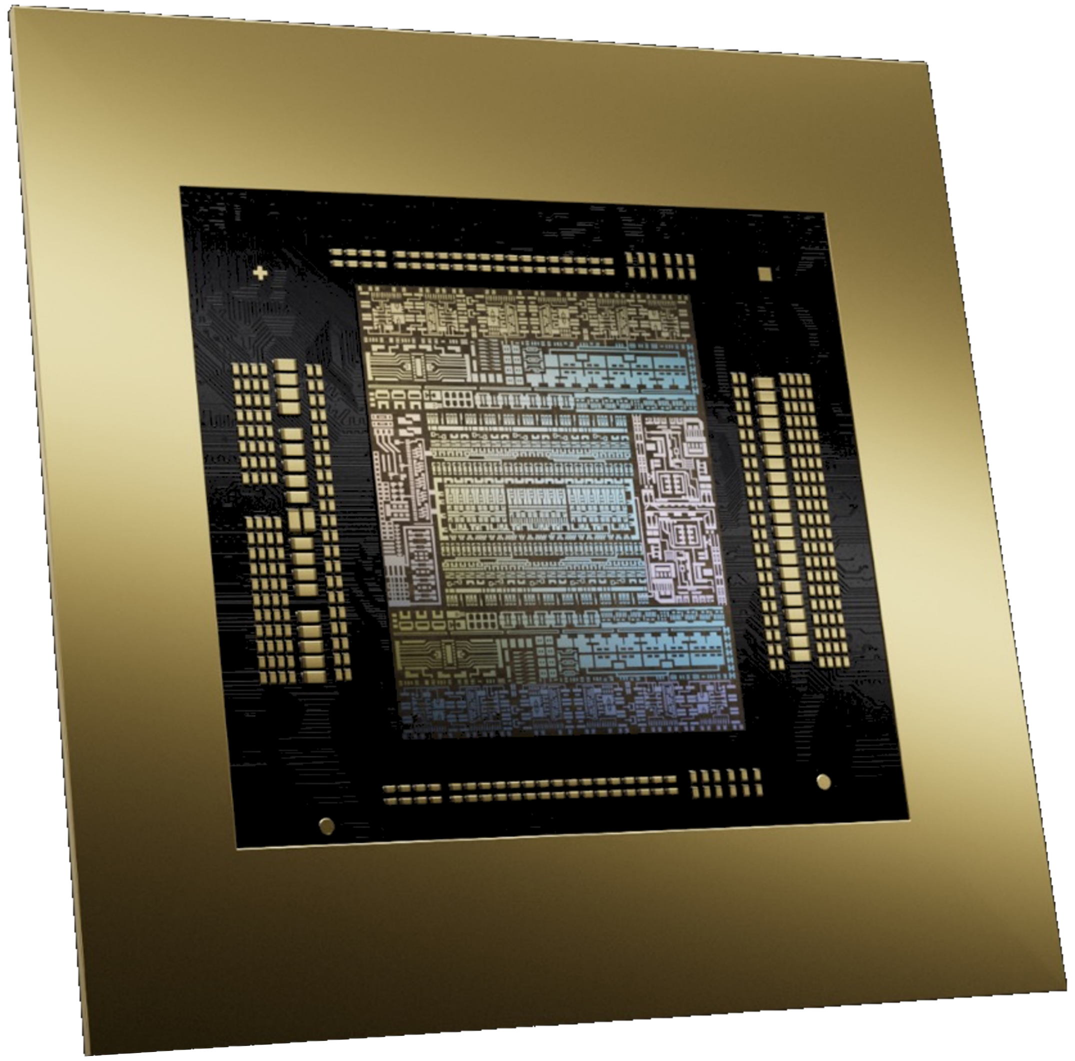


DGX B200

- Next generation DGX system with 8X NVIDIA Blackwell GPUs
- 1.4TB of GPU memory, enabling training of large generative AI models
- Purpose-built, unified platform for every workload from training, to fine-tuning, to inference
- Delivers 3X AI training and 15X AI inference performance as previous generation (DGX H100)
- Latest Blackwell architecture in a scalable, air-cooled design

Announcing Fifth Generation NVLink and NVLink Switch Chip

Efficient Scaling for Trillion Parameter Models



7.2 TB/s Full all-to-all Bidirectional Bandwidth

Sharp v4 plus FP8

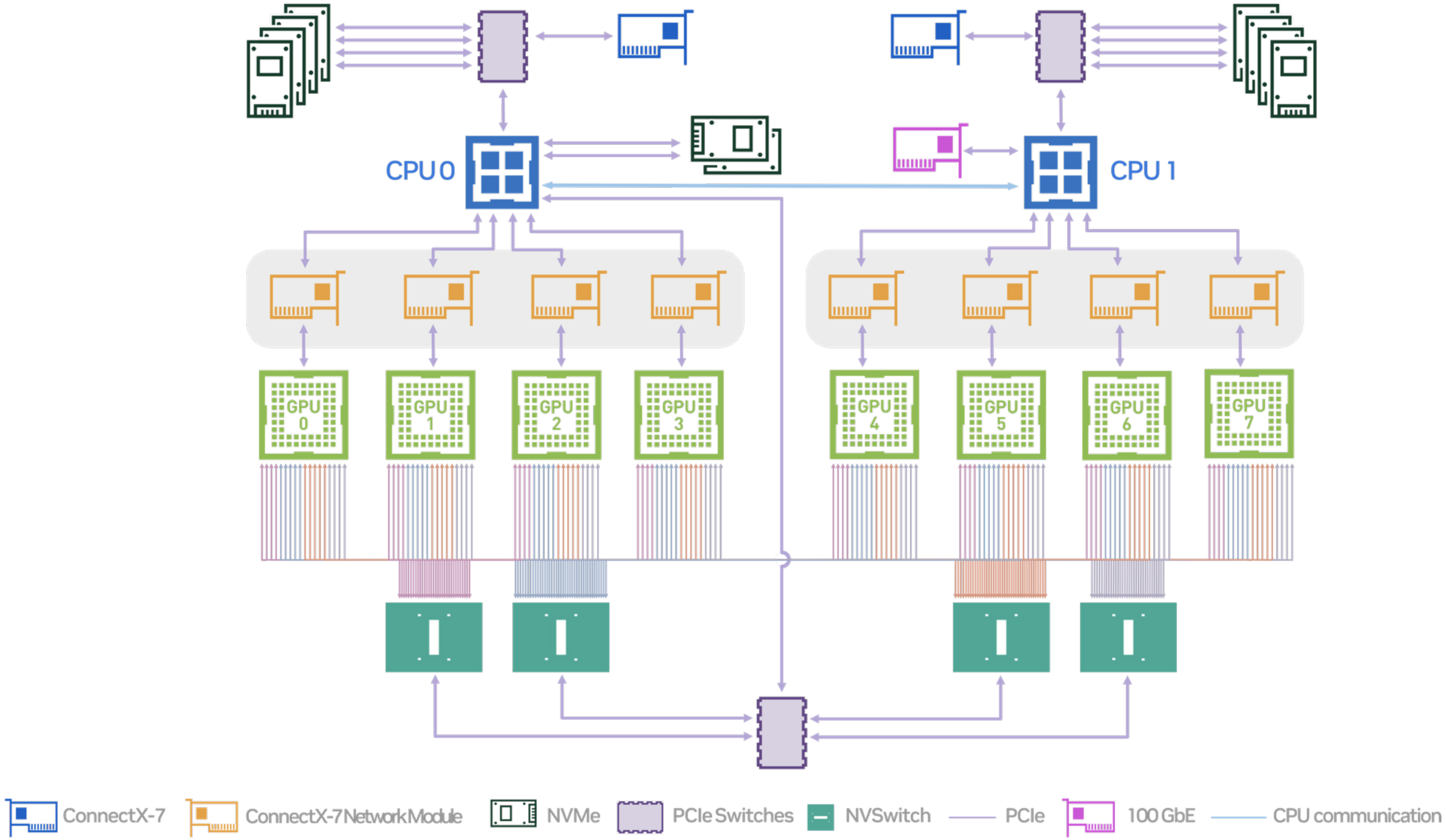
3.6 TF In-Network Compute

Expanding NVLink up to 576 GPU NVLink Domain

18X Faster than Today’s Multi-Node Interconnect

SERVER DESIGN

Facilitating for Various Forms of Parallelism



LLAMA 2 TRAINING TIME

Hypothetical Training Time on single NVIDIA A100 GPUs

Single GPU

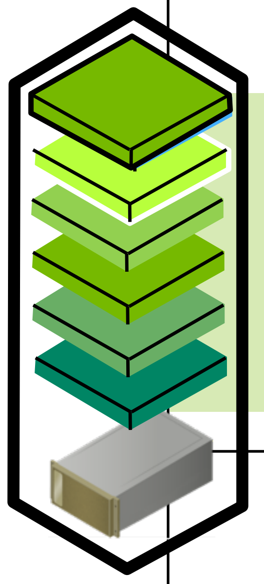
meta-llama

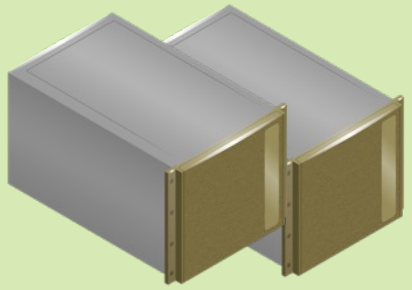
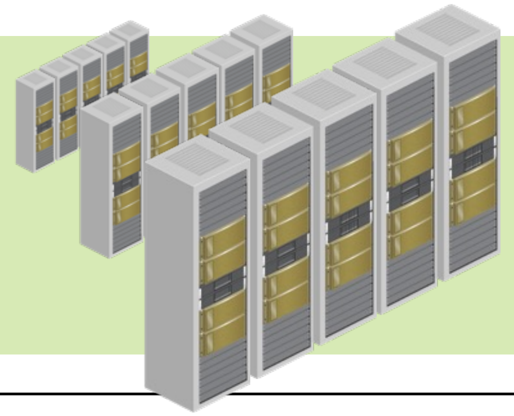
24 years +

Beyond a single server

Powering Your AI Journey End-to-End

Delivering incremental value for your DGX data center, as your needs grow



	Day One your 1st DGX systems	Scaled Infrastructure DGX BasePOD	AI CoE DGX SuperPOD
DGX Systems powers every step in your AI journey			

Base Command capabilities that help you grow your AI platform

- NVIDIA AI Enterprise:
  - Pre-trained models, optimized frameworks
  - Accelerate data prep tasks
  - Customize/fine-tune pre-trained models
  - Optimize/accelerate inference
- Kubernetes or Slurm scheduling
- Add/manage DGX within your existing compute infrastructure (cloud, non-GPU)
- Run Jupyter notebooks
- Cloud-based private registry
- Accelerate storage & network IO
- Fully optimized OS stack
- DGX-Ready Software for AI workflow management and MLOps (Optional)

Continuous roadmap of innovative features delivered to customers

DGX Cloud - The Cloud-First Way to Get Access to Your Own DGX AI Supercomputer

DGX SUPERPOD

Modular Architecture

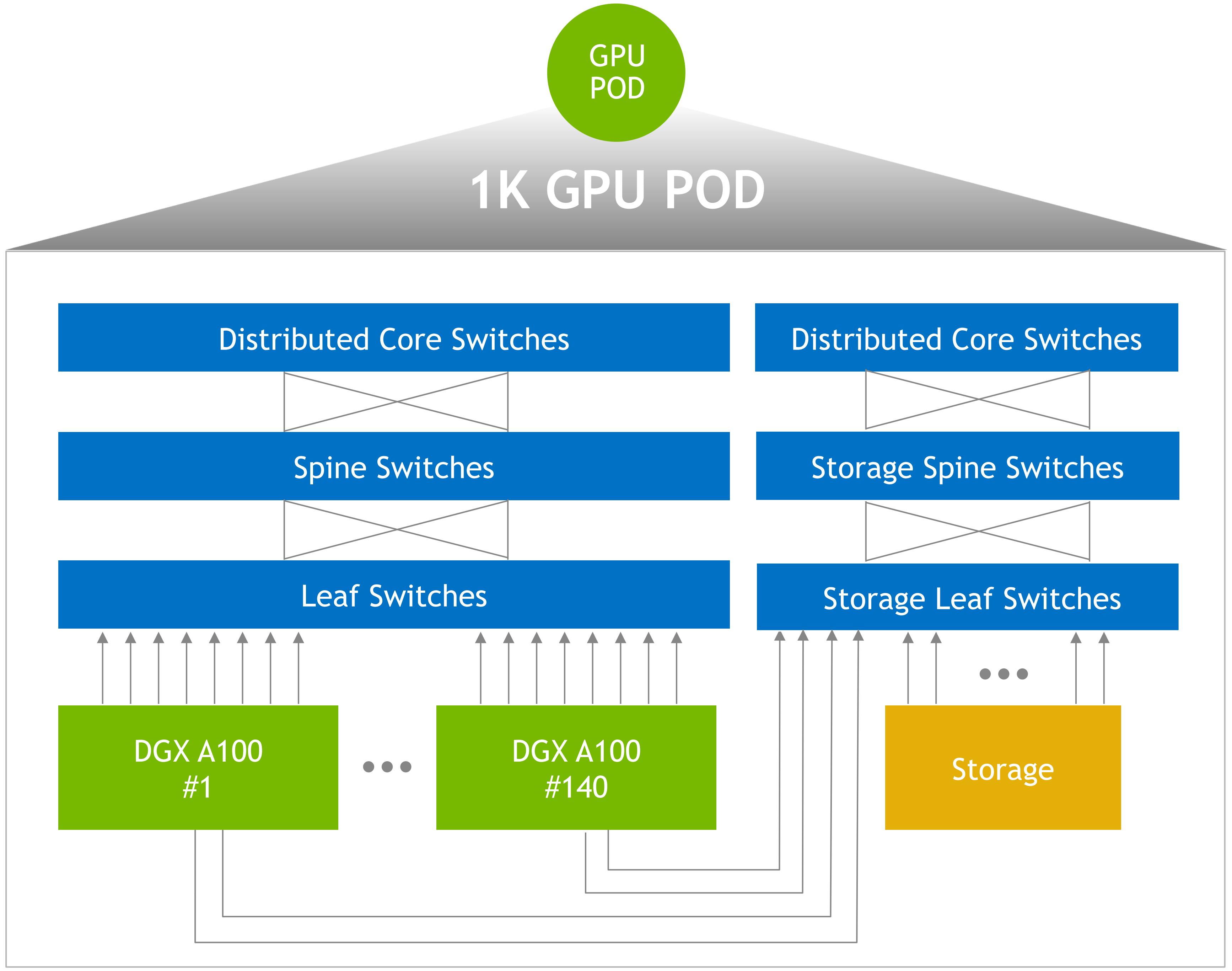
- 140 DGX A100 nodes (1,120 GPUs) in a GPU POD
- 1st tier fast storage - DDN AI400x with Lustre
- Mellanox HDR 200Gb/s InfiniBand - Full Fat-tree
- Network optimized for AI and HPC

DGX A100 Nodes

- 2x AMD 7742 EPYC CPUs + 8x A100 GPUs
- NVLINK 3.0 Fully Connected Switch
- 8 Compute + 2 Storage HDR IB Ports

A Fast Interconnect

- Modular IB Fat-tree
- Separate network for Compute vs Storage
- Adaptive routing and SharpV2 support for offload

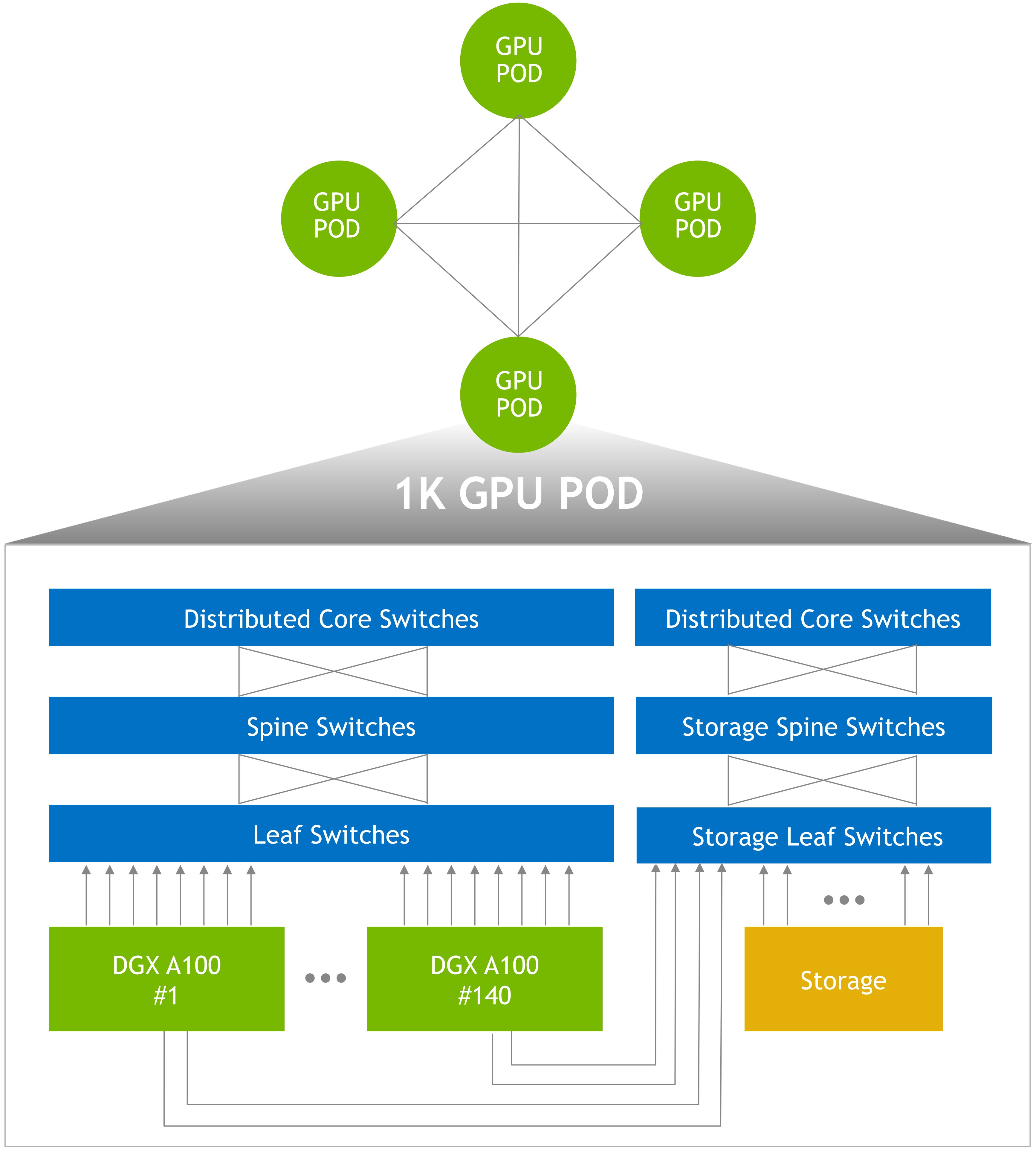


DGX SUPERPOD

Extensible Architecture

POD to POD

- Modular IB Fat-tree or DragonFly+
  - Core IB Switches Distributed Between PODs
  - Direct connect POD to POD



The New GPU

Announcing GB200 NVL72

Delivers New Unit of Compute

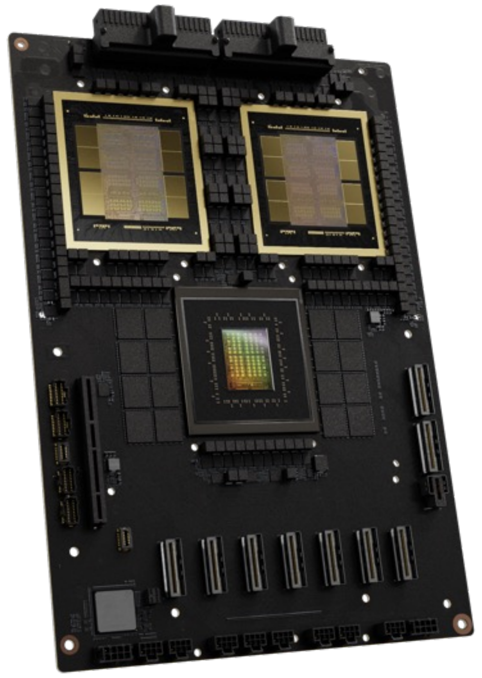
GB200 NVL72

- 36 GRACE CPUs
- 72 BLACKWELL GPUs
- Fully Connected NVLink Switch Rack

Training FP8	720 PFLOPs
Inference FP4	1,440 PFLOPs
NVL Model Size	27T params
Multi-Node All-to-All	130 TB/s
Multi-Node All-Reduce	260 TB/s

GB200 NVL72 Compute and Interconnect Nodes

Building Blocks for the GB200 NVL72 Rack



GB200 SUPERCHIP

40 PETAFLOPS FP4 AI INFERENCE
20 PETAFLOPS FP8 AI TRAINING
864GB FAST MEMORY

GB200 SUPERCHIP COMPUTE TRAY

2x GB200
80 PETAFLOPS FP4 AI INFERENCE
40 PETAFLOPS FP8 AI TRAINING
1728 GB FAST MEMORY
1U Liquid Cooled
18 Per Rack

NVLINK SWITCH TRAY

2x NVLINK SWITCH CHIP
14.4 TB/s Total Bandwidth
SHARPv4 FP64/32/16/8
1U Liquid Cooled
9 Per Rack

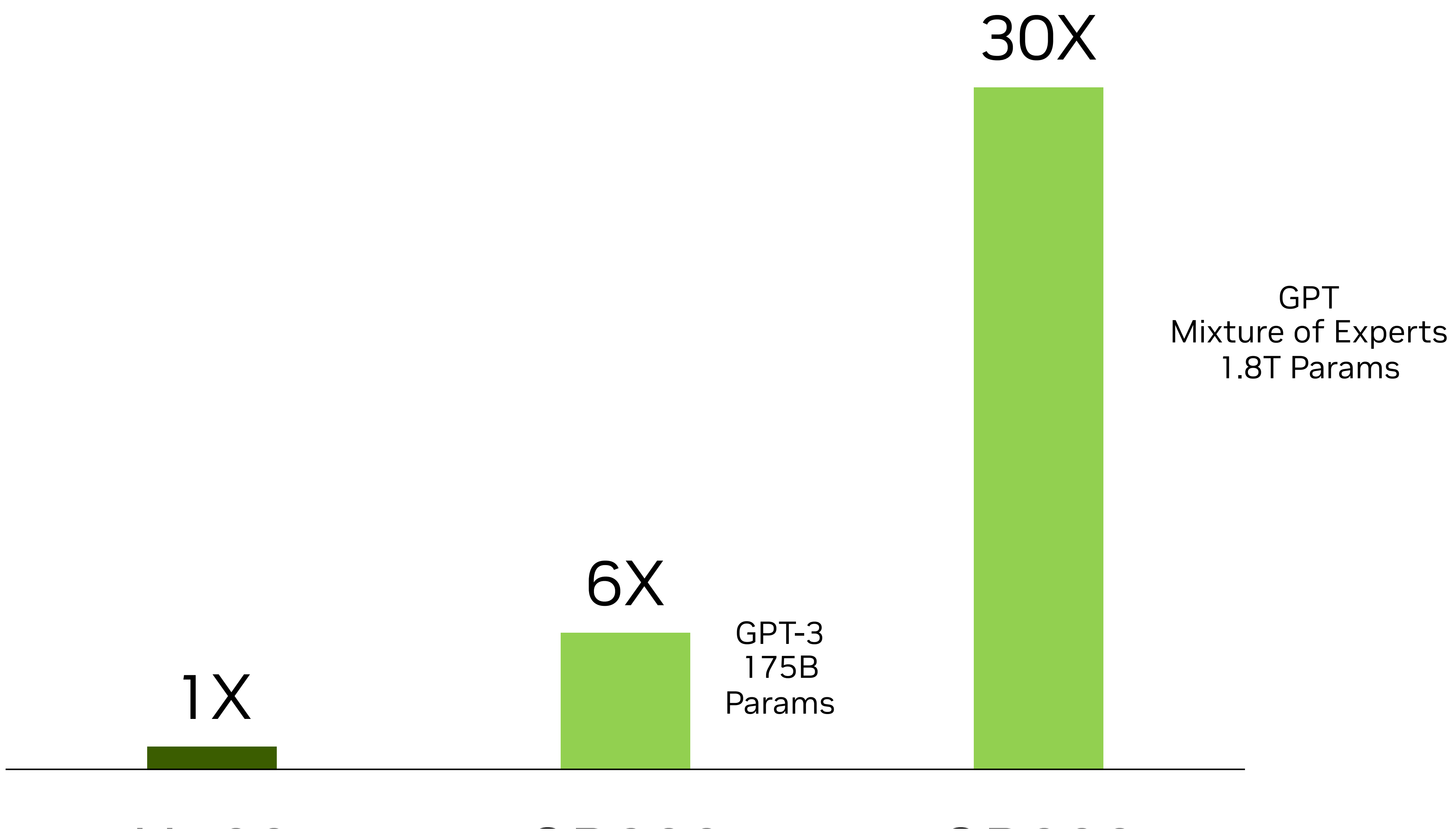
GB200 NVL72 Enabling Trillion Parameter AI

30x Realtime Inference Mixture of Experts Inference, 25X Improved Energy Efficiency

30X
Higher Throughput

25X
Energy Efficiency

25X
Lower TCO



Projected performance subject to change
Token-to-token latency (TTL) = 50 milliseconds (ms) real time

Blackwell for Every Generative AI Use Case

Delivering the New Era of Performance for Every Data Center

GB200 NVL72

Compute for Trillion Parameter Scale AI
Maximum Performance and Lowest TCO

HGX B200

Best Performance and TCO for HGX Platform

HGX B100

Drop-in Upgrade for Existing Hopper Infrastructure

Blackwell Ecosystem

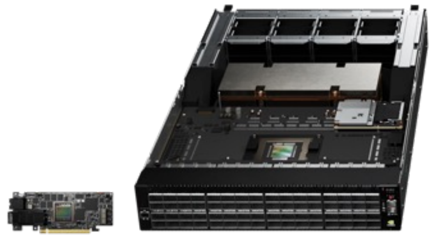
Coming Later 2024

aws Google Cloud Microsoft Azure ORACLE CLOUD Infrastructure

Scaleway CoreWeave NEXGEN CLOUD indosat OOREDOO HUTCHISON Lambda YOTTA NORTHERN DATA GROUP IBM Cloud APPLIED DIGITAL CRUSOE Singtel

DELL Technologies EVIDEN Hewlett Packard Enterprise CISCO SUPERMICRO FUJITSU GIGABYTE ASUS SoftBank

FOXCONN HON HAI TECHNOLOGY GROUP PEGATRON Inventec AIVRES wiwynn wistron ASRock Rack QCT Systems



Spectrum-X800

Quantum-X800

GB200 NVL72

HGX B200

HGX B100

I do not care about training!
What about inference?

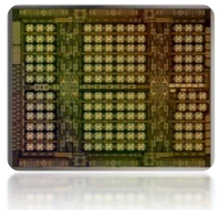
NVIDIA MGX

Scientific Computing | Data Processing | LLM Training | Gen AI Inference | Cloud Video & Graphics | EDA SDA CADD | Enterprise Gen AI | Edge AI

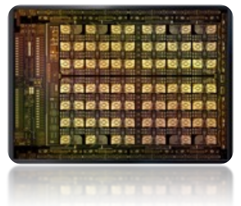
$1T Global Datacenter Infrastructure transitioning to accelerated computing and generative AI

Accelerated Computing

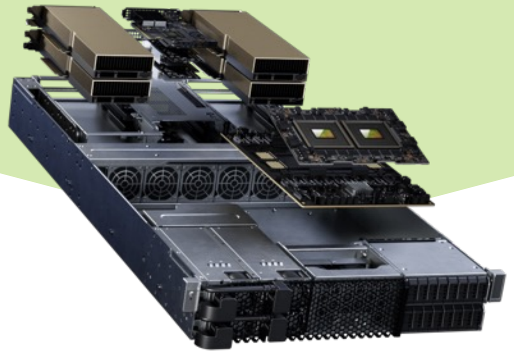
GPU



CPU



DPU



A Modular Reference Architecture for Accelerated Computing

Time-to-Market

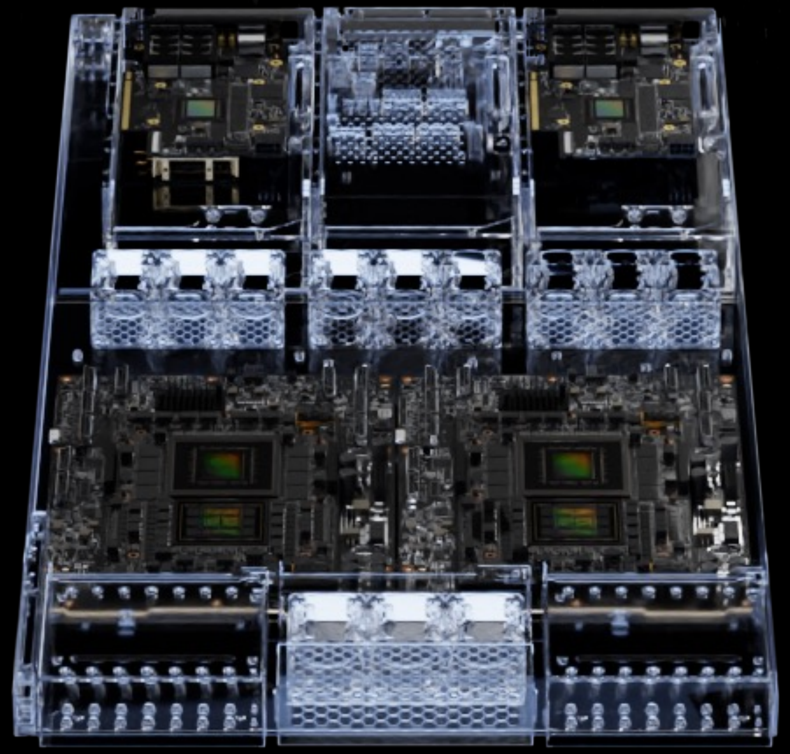
Multi-Gen Compatibility

Open and Flexible

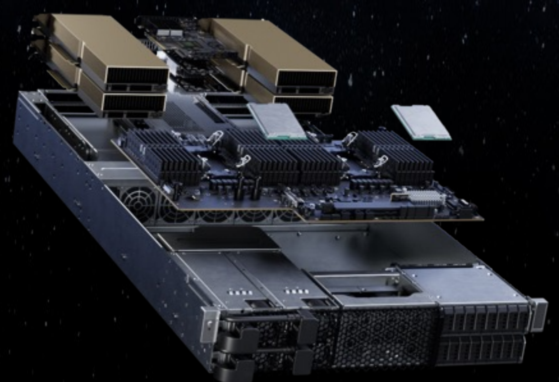
MGX – Modular Reference Designs - To Enable Large Number of Configurations

New!

NVLink Dual GH200 system

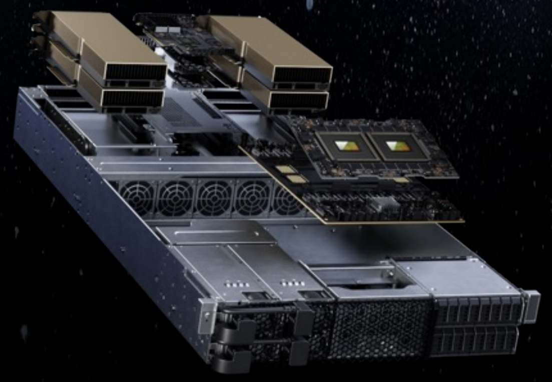


144 Core Grace CPU | 8 PFLOPS Hopper GPU
288 GB HBM3e | 10 TB/s



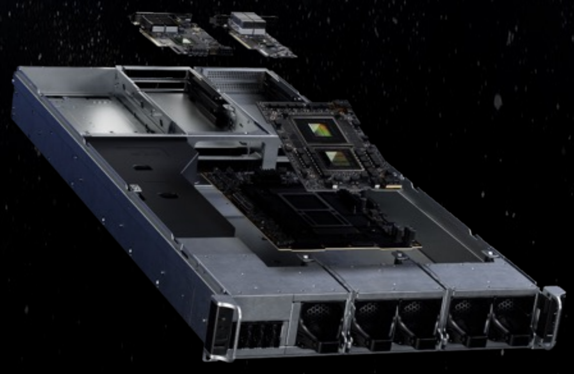
2U X86 Mainstream Server

2U | x86 | 4 L40 | BF-3 | 2 CX-7 | 6 PCIE



2U Grace Mainstream Server

2U | Grace | 4 L40 | BF-3 | 2 CX-7 | 6 PCIE

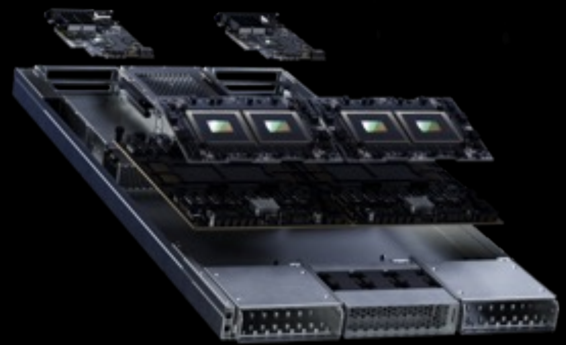


Grace Hopper Server

2U | Grace-Hopper | BF-3 | CX-7 | 6 PCIE

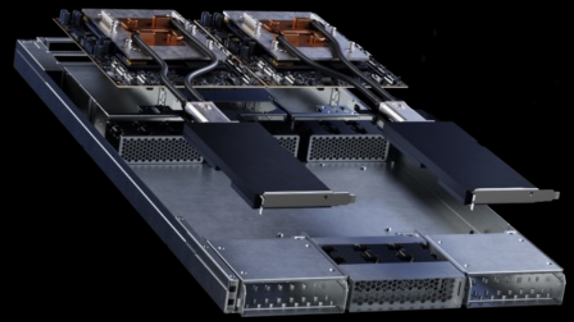
Grace-Hopper Aerial Server

1U | Grace-Hopper | 2 BF-3 | 4 PCIE



Dense General-Purpose Grace CPU Server

1U | 2 Grace | 2 BF-3 | 4 PCIE

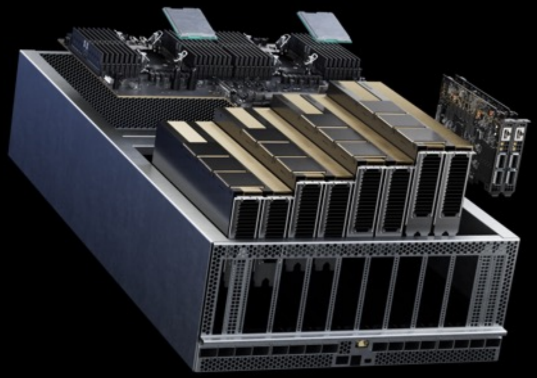


Grace-Hopper Liquid-Cooled Server for HPC

1U | 2 Grace-Hopper | 2 BF-3 | 4 PCIE

Grace Cloud Gaming Server

2U | Grace | 10 L4 | BF-3 | 11 PCIE



Hopper NVL Inference Server

4U | x86 | 8 H100 NVL | 2 BF-3 | 10 PCIE

Grace-Hopper Aerial Server Short Depth

2U 450mm | Grace-Hopper | BF-3 | CX-7

ABOUT ME

Adam Grzywaczewski – adamg@nvidia.com