Compute Trends Across Three Eras of Machine Learning

Jaime Sevilla*†, Lennart Heim*§, Anson Ho*, Tamay Besiroglu*‡, Marius Hobbhahn *¶, Pablo Villalobos*

2022 International Joint Conference on Neural Networks (IJCNN) | 978-1-7281-8671-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCNN55064.2022.9891914

Abstract— **Compute, data, and algorithmic advances are the three fundamental factors that drive progress in modern Machine Learning (ML). In this paper we study trends in the most readily quantified factor – compute. We make three novel contributions: (1) we curate a dataset with the training compute of 123 milestone ML systems, 3× larger than previous such datasets. (2) We frame the trends in compute in in three eras – the Pre Deep Learning Era, the Deep Learning Era, and the Large-Scale Era, based on our identification of a novel trend emerging around 2015. (3) We find a Deep Learning Era compute doubling time of around 6 months, significantly longer than previous findings. Overall, our work highlights the fast-growing compute requirements for training advanced ML systems.**

Index Terms—**machine learning, artificial intelligence, deep learning, computational efficiency, AI accelerators, backpropagation, high performance computing**

I. Introduction

The field of Machine Learning (ML) has been progressing rapidly over the last decade, with significant implications for industry, policy, and society. These developments have been driven by advances in AI hardware, increased data availability, and algorithmic improvements, among others. However, quantifying these driving factors is challenging, and as such our present understanding of their relative importance is largely qualitative or limited by insufficient data.

Our investigation takes a major step in rectifying this, where we focus on a relatively quantifiable factor – *the total training compute over the final training run of a ML experiment*, measured in floating-point operations (FLOPs).[1] Following the example of [1, 2, 3], we adopt the more colloquial term, *compute*, to refer to this factor.

This paper is a detailed investigation into the compute demand of milestone ML models over time, with the following contributions:

1) **We curate a dataset of 123 milestone ML systems** annotated with training compute, 3× larger than previous such datasets
2) **We frame the trends in compute in terms of three distinct eras**: the **Pre Deep Learning Era**, the **Deep Learning Era** and the **Large-Scale Era**
3) We calculate compute doubling times during each era, and **find that previously-obtained doubling times during the Deep Learning Era were significantly overstated**

Both our dataset, figures, and an interactive visualization are publicly available.

II. Related work

Our work significantly builds upon prior study into compute trends, most notably the *AI and Compute* investigation by Amodei and Hernandez [4], and the subsequent addendum by Sastry et al. [5]. These investigations identified two eras of compute growth and a rapid 3.4 month doubling time between 2012 and 2018, which Lyzhov [6] argues is not predictive of progress post-2018.[2]

The study of scaling laws [3, 9, 10, 11, 12, 13] relate these compute trends to model performance, and are actively used by large corporations to inform their training resource requirements [2, 14]. This has raised interests in projecting future compute requirements and hardware progress [15, 16], and hints at the growing importance of gathering compute data and investigating trends. This is being done by several initiatives [17, 18, 19, 20], the data from which we use (with permission) to inform our own work.

Compared to prior work, our data collection is significantly more comprehensive, containing 3× more ML models than previous ones and including data up to 2022. We also offer novel interpretations of previous data, which we believe have important implications for understanding progress in ML.

III. Methods

A. Model selection

Models in our dataset are only chosen from papers containing an explicit Machine Learning component with experimental results, and papers describing attempts to advance the state-of-the-art (SotA). Moreover, we believe it to be most informative to study milestone papers at the frontiers of ML progress, thus we further require at least one notability criterion. Specifically, the paper must have received over 1000 citations, be of clear historical importance, demonstrate an important SotA advance, or be deployed in a notable context.

We curated these **milestone systems** from literature reviews, Papers with Code, historical accounts, previous datasets, most cited publications of top conferences, and suggestions from individuals.

Our selection is inevitably somewhat subjective, although we believe our dataset to be a robust collection of models that

*Epoch, †University of Aberdeen, ‡MIT Computer Science & Artificial Intelligence Laboratory, §Centre for the Governance of AI, ¶University of Tübingen

[1] In the literature this is also commonly referred to as FLOP. We distinguish FLOPs from floating point operations *per second*, which we instead denote as FLOP/s.

[2] For example, the most compute-intensive model of 2020 (GPT-3) [7] only requiring 1.5× more compute for training than the most compute-intensive model of 2017 (AlphaGo Zero) [8].

978-1-7281-8671-9/22/$31.00 ©2022 IEEE

is representative of how the ML SotA has evolved over the past seven decades. The notability criteria are especially hard to assess for recently published models (e.g. 2020); we use a more subjective assessment for these systems.

B. Estimating training compute

For the vast majority of models that met the selection criteria, the training compute in FLOP/s was not directly mentioned in the associated paper. We thus used the two estimation techniques introduced by Amodei and Hernandez [4], and lean heavily on our previous analysis on the validity of these methods and how best to apply them [21]. We summarise the two methods and our findings below:

1) **Architecture-based**: Counting the number of operations based on the model architecture, and making appropriate assumptions about the ratio of FLOP/s in a forward pass relative to a backward pass through the model. Specifically, the training compute C is given by

$$C = 2 \times N_C \times R \times D \times E, \tag{1}$$

where N_C is the number of connections[3], $R = \frac{\text{Operations per backward pass}}{\text{Operations per forward pass}}$ is the backward-forward FLOP ratio, D is the number of training examples, and E is the number of training epochs. Empirically, we find $R = 2$ to be a fairly good approximation [21], and use this as a default.

2) **Hardware-based**: Using the hardware details and information about usage during training to estimate compute. If T is the time for the final training run, n is the number of chips of a particular type of hardware, P is the peak FLOP/s of the chip, and U is the utilization rate, we have:

$$C = T \times n \times P \times U. \tag{2}$$

We expect U to depend heavily on the training setup, and our previous experiments support this view [21]. Depending on whether the paper was published at a large research corporation and the publication year, we generally default to $U \approx 30\%$.

We lean heavily on our previous work demonstrating the validity of the assumptions used in these estimation techniques [21].

The hardware-based approach was often more feasible for sophisticated architectures, and is the dominant method we use for models published after 2017.

This data was gathered by manually searching publications for the required architecture and hardware details, which we used to estimate the total number of FLOPs for the final training run. A surprisingly large fraction of papers did not provide sufficient information to apply either estimation method, and we found it necessary to contact the authors or impute hardware information based on the publication year.[4] Our reasoning for each estimate is annotated in the respective cell of the main dataset.

Note that these calculations only help determine the compute for the *final training run* of ML models. While ML systems are often trained multiple times, and significant compute usage goes towards experimentation, information regarding this is typically not accessible from papers. Therefore, our dataset only accounts for compute used for the final training run, and our reported compute may not be representative of the monetary costs of the full experiment.

We check the consistency of these two methods by considering a random selection of papers, and find that they yield estimates that are within a factor of 2 of each other.

C. Analysing compute trends

The reported regressions and doubling rates are derived from log-linear fits to the training compute. Where confidence intervals are indicated, those are derived from a bootstrap with $B = 1000$ samples. To account for the uncertainty of our estimates, we randomly adjust each estimate by randomly multiplying it by a number between $\frac{1}{2}$ and 2 (where the factor of 2 is derived from the aforementioned range of empirical differences when comparing the two compute estimation methods [21]). The concrete distribution we sample the random adjustment from is log uniform between $\frac{1}{2}$ and 2. We use the notation [quantile 0.025; median; quantile 0.975] to indicate 95% confidence intervals.

Throughout the article, we have excluded low-compute outliers from the dataset, since we are actively interested in studying high compute models that are pushing the boundaries of ML. This is done by calculating the log training compute Z-score of each model with respect to other models whose publication date is within 1.5 years. We exclude models whose Z-score is 2 standard deviations below the mean.

This criteria results in the exclusion of 5 models out of 123 between 1952 and 2022. The models excluded this way are often from relatively special domains, such as poker, board games, and hide and seek.

Later we used a similar methodology to automatically select papers with exceedingly high compute, choosing papers that exceed the $Z > 0.76$ threshold after 2016. In both cases, we first decided by visual inspection which papers to mark as outliers and then chose the thresholds accordingly to automatically select them.

IV. Discussion

We interpret our data in terms of three distinct eras defined by two transition points:

1) **Pre Deep Learning Era**: From 1950 to around 2010, the training compute doubles every 17 to 29 months. This is roughly inline with Moore's Law, according to which

[3]This is not the same as the number of parameters; rather it is the number of connections between neurons in an *unrolled* neural network.

[4]For instance, if the hardware accelerator was not specified, we estimated the peak FLOP/s by using the average peak performance of commonly used hardware accelerators of publications in the same year. In addition, we generally assumed that models were trained with a dominant floating point representation in 32bit (FP32), unless there is strong evidence to the contrary (e.g. large corporations tend to use FP16 from 2020 onward).

Period	Data	Scale (start to end)	Slope	Doubling time
1952 to 2010 Pre Deep Learning Era	No low outliers ($n = 19$)	3e+04 to 2e+14 FLOPs	0.2 OOMs/year [0.1; 0.2; 0.2]	21.3 months [17.0; 21.2; 29.3]
2010 to 2022 Deep Learning Era	No outliers ($n = 80$)	7e+14 to 2e+18 FLOPs	0.6 OOMs/year [0.4; 0.7; 0.9]	5.7 months [4.3; 5.6; 9.0]
September 2015 to 2022 Large-Scale Era	High outliers ($n = 19$)	4e+21 to 8e+23 FLOPs	0.4 OOMs/year [0.2; 0.4; 0.5]	9.9 months [7.7; 10.1; 17.1]

TABLE I: Summary of our main results. In 2010 the trend accelerated along the with the popularity of Deep Learning, and in late 2015 a new trend of large-scale models emerged.

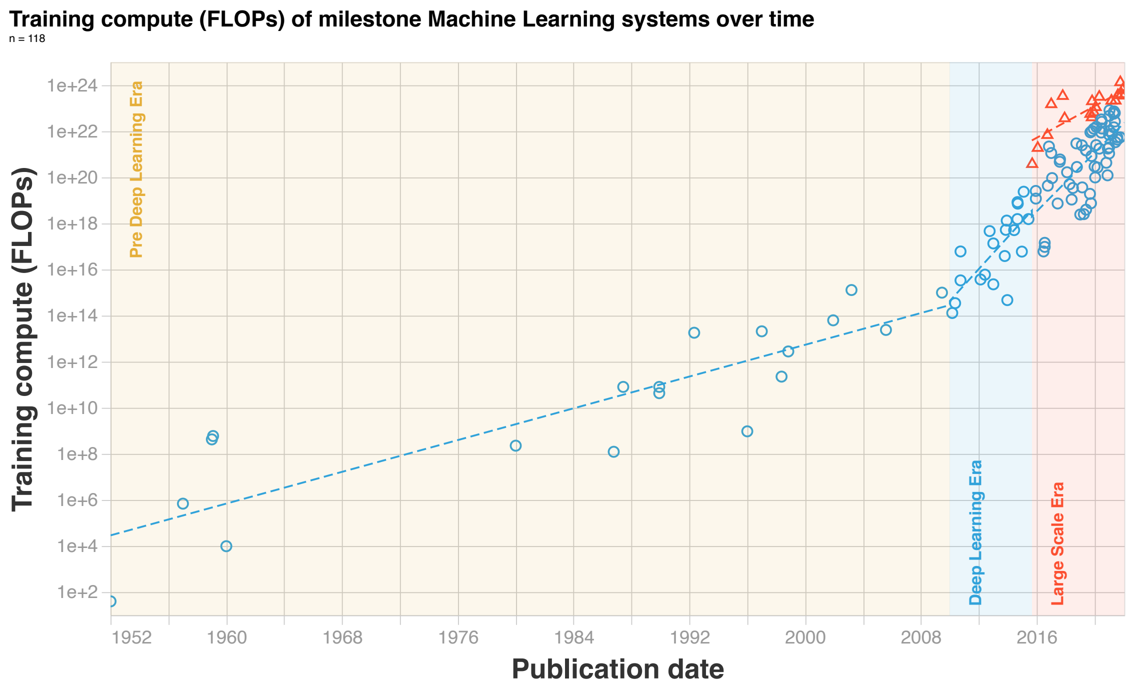

Fig. 1: Trends in $n = 118$ milestone ML models between 1952 and 2022. We distinguish three eras. Notice the change of slope circa 2010, matching the advent of Deep Learning; and the emergence of a new large-scale trend in late 2015.

transistor density doubles roughly every two years [22] (this is often simplified to computational performance doubling every two years)

2) **Deep Learning Era**: After around 2010, we observe a slope discontinuity where the compute doubles every 4 to 9 months, significantly longer than the results obtained in [4]
3) **Large-Scale Era**: We argue that a new trend of large-scale models, with compute significantly higher than other models published in the same year, emerges in 2015 with the release of AlphaGo [8]. This grows at a slower rate than the Deep Learning trend, doubling roughly every 8 to 17 months

The data arguably lies along three log-linear trends – one corresponding to the Pre Deep Learning Era (1952 to 2010), a second corresponding to regular (i.e. not large-scale) models after the advent of Deep Learning (2010 to 2022), and a large-scale trend from 2015 to 2022.

A. When did the Deep Learning Era start?

One potential source of error is the ambiguity in the transition points - for instance, in our choice of the start of the Deep Learning Era. In particular, our data (as shown in Figure 1) does not allow for resolution of the transition to Deep Learning at the level of a year.

Many authors decide to start the Deep Learning Era with the release of AlexNet in 2012 [4, 23], but there is some room for debate regarding this, and we instead believe that 2010 is most inline with the available evidence:

- Many models preceding AlexNet have features associated with Deep Learning, including model size and depth [24, 25, 26, 27, 28], GPU-based training [25, 29, 30, 31, 32], and better performance than traditional ML approaches [26, 27, 28, 31]

Period	Outliers	Scale (FLOPs)	Slope	Doubling time	R²
1952-2009	All models ($n = 19$)	3e+04 / 2e+14	0.2 OOMs/year [0.1; 0.2; 0.2]	21.3 months [16.2; 21.3; 31.3]	0.77
1952-2011	All models ($n = 26$)	1e+04 / 3e+15	0.2 OOMs/year [0.1; 0.2; 0.2]	19.6 months [15.6; 19.4; 25.0]	0.83
2010-2022	All models ($n = 98$)	1e+15 / 6e+22	0.7 OOMs/year [0.6; 0.7; 0.7]	5.6 months [5.0; 5.6; 6.2]	0.70
	Regular-scale ($n = 77$)	4e+14 / 2e+22	0.7 OOMs/year [0.6; 0.7; 0.7]	5.6 months [5.1; 5.6; 6.2]	0.78
2012-2022	All models ($n = 91$)	1e+17 / 6e+22	0.6 OOMs/year [0.5; 0.6; 0.7]	5.7 months [4.9; 5.7; 6.7]	0.58
	Regular-scale ($n = 80$)	4e+16 / 2e+22	0.6 OOMs/year [0.5; 0.6; 0.7]	5.7 months [4.9; 5.7; 6.7]	0.69

TABLE II: Log-linear regression results for ML models from 1952 to 2022.

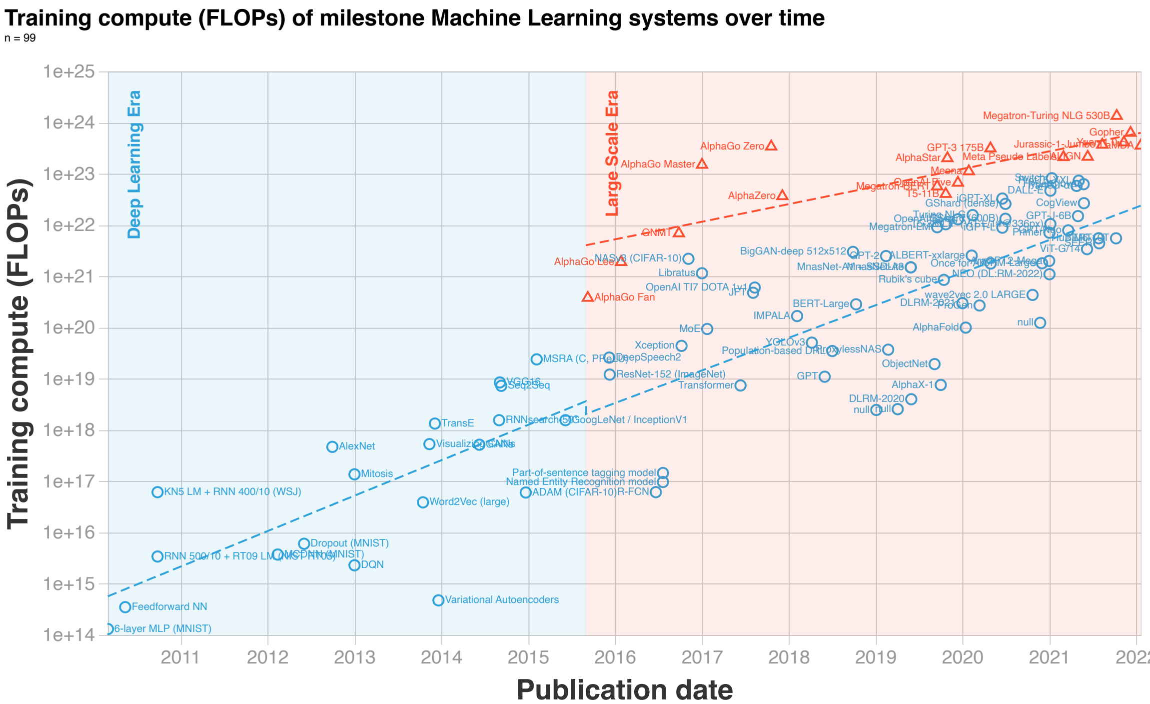

Fig. 2: Trends in training compute of $n = 99$ milestone ML systems between 2010 and 2022. Notice the emergence of a possible new trend of large-scale models around 2016. The trend in the remaining models stays the same before and after 2016.

- There is evidence that between 2009 and 2012 (prior to the 2012 ImageNet competition won by AlexNet), the field of speech recognition realised that Deep Learning would be capable of achieving major breakthroughs in the domain. In particular, Deng, Yu, and Hinton suggest that "deep architectures with efficient learning algorithms" would be needed to overcome challenges [33], and [34] suggests that leading Speech Recognition researchers held a shared vision of Deep Learning driving major advances in their field

Hence we argue that 2010 is the starting date most consistent with the evidence, because: (a) the use of GPUs to train large ML models was already common at the time, (b) there were at least a few Deep Neural Networks that achieved highly competitive performance (notably [26, 27, 31]), and (c) this timeline is consistent with the adoption of Deep Learning in Speech Recognition. Although we use 2010 as the default start of the Deep Learning Era, our conclusions remain unchanged if 2012 is used instead (see Table II).

B. Trends in the Large-Scale era

The second transition point is more speculative – our data suggests the emergence of a new trend of large-scale models, starting with AlphaGo in late 2015 and continuing to the present (see Figure 2). This represents a bifurcation of the Deep Learning trend that persisted from 2010 to 2015, with the trend of regular-scale models continuing unperturbed post-2016.

We believe that there are several arguments in favour of this framing of large-scale trends:

- **The trend of regular-scale Deep Learning models continues unperturbed post-2016**, doubling every 5 to

Period	Data	Scale (FLOPs)	Slope	Doubling time	R^2
2010-2016	All models ($n = 20$)	6e+14 / 3e+18	0.7 OOMs/year [0.4; 0.6; 0.9]	5.3 months [3.9; 5.2; 8.5]	0.55
2016-2022	All models ($n = 79$)	1e+19 / 5e+22	0.5 OOMs/year [0.4; 0.6; 0.8]	6.7 months [4.9; 6.6; 10.0]	0.33
	Regular scale ($n = 60$)	3e+18 / 2e+22	0.6 OOMs/year [0.5; 0.6; 0.8]	5.9 months [4.4; 5.8; 7.9]	0.48
	Large-Scale ($n = 19$)	4e+21 / 6e+23	0.3 OOMs/year [0.1; 0.3; 0.5]	10.7 months [7.9; 10.6; 25.6]	0.66

TABLE III: Results of a log-linear regression for data between 2010 and 2022. The trend of regular-scale models before 2015 continues uninterrupted afterwards.

Period	Data	Scale (FLOPs)	Slope	Doubling time	R^2
2016-2022	Regular-scale models $Z < 0.76$, ($n = 63$)	3e+18 / 1e+22	0.6 OOMs/year [0.4; 0.6; 0.8]	6.0 months [4.6; 6.0; 8.5]	0.46
	Large-scale models $Z > 0.76$, ($n = 20$)	3e+21 / 6e+23	0.4 OOMs/year [0.2; 0.4; 0.5]	10.3 months [7.6; 10.4; 21.9]	0.63
	Regular-scale models $Z < 0.6$, ($n = 57$)	3e+18 / 9e+21	0.6 OOMs/year [0.4; 0.6; 0.8]	6.0 months [4.6; 6.1; 8.6]	0.48
	Large-scale models $Z > 0.6$, ($n = 26$)	3e+21 / 4e+23	0.3 OOMs/year [0.2; 0.3; 0.5]	10.7 months [7.8; 10.9; 19.5]	0.57
	Regular-scale models $Z < 0.54$, ($n = 51$)	3e+18 / 5e+21	0.5 OOMs/year [0.4; 0.6; 0.7]	6.7 months [4.9; 6.7; 9.9]	0.45
	Large-scale models $Z > 0.54$, ($n = 32$)	2e+21 / 3e+23	0.3 OOMs/year [0.2; 0.3; 0.4]	11.6 months [8.3; 11.6; 22.1]	0.49

TABLE IV: Results from varying different thresholds for the large-scale trend. Our results remain largely consistent regardless of the chosen Z value.

6 months (see Table III). In contrast, the large-scale trend follows a significantly shorter doubling time, at 9 to 10 months, supporting the separate categorisation of large-scale models

- **The large-scale trend explains the increased emphasis on resource-intensive projects**. Notably, we find that all 19 models in the large-scale trend were almost exclusively published by industry corporations, and 17 were published by DeepMind, Google AI, OpenAI, or Microsoft. In contrast, for models following the regular-scale trend in the Large-Scale Era, roughly 80% were published by industry. These large organisations presumably have larger training budgets[5], enabling them to achieve a drastic departure in funding, resulting in a novel trend.
- **The large-scale trend is better at predicting developments post-2017**. An alternative interpretation of our data is to consider a single trend, showing a 4 month doubling time from September 2012 to December 2017, and slowing to a 5 month doubling time afterward. However, [6] points out that this explanation does not extend past 2017, since the doubling time is excessively short. In comparison, the large-scale trend (with a doubling time of 9-10 months) better accounts for developments after 2017.

Note that it is difficult to rule out the possibility that the large-scale trend is due to noise, given the current lack of data. We nevertheless believe that there is good evidence in favour of our hypothesised trend being true.

Another source of uncertainty is in our selection criteria for large-scale models. There is a reasonable case for including NASv3, Libratus, Megatron-LM, T5-3B, OpenAI Five, Turing NLG, iGPT-XL, GShard (dense), Switch, DALL-E, Pangu-α, ProtT5-XXL and HyperClova on either side of the division. In Table IV we show the effects of choosing different Z-value thresholds to separate the Large-Scale models – overall, the differences are small, and our overarching conclusions remained unchanged.

C. Comparison with previous work

Our results contrast with [4], who find a much faster doubling period of 3.4 months between 2012 and 2018, and with [6], who finds a much longer doubling period of >2 years between

[5]For instance, AlphaGo Zero in 2017 [8] is estimated to have cost $35M [35] and AlphaStar [36] following in 2019 with an estimated cost of $12M [37]. GPT-3 [7], a recent SotA NLP model, has been estimated to have cost around $4.6M to train [38]. We do not know the exact spending of the relevant companies and these should be treated as rough estimates.

2018 and 2020. We make sense of these discrepancies by noting that their analyses have significantly fewer data samples and assume a single trend, whereas ours studies large-scale and regular-scale models separately. Furthermore, the evidence of the large-scale trend only emerged recently, such that previous analyses would have been unable to distinguish large-scale and regular-scale trends.

D. Limitations

Note that selection effects are unavoidable due to the notability criteria. Our model search is further biased towards models that are found in academic publications (as opposed to closed-source commercial systems) written in English. Although we believe these do not believe these biases change our conclusions, we nevertheless urge caution when jumping to strong conclusions from our data.

V. Conclusion

In this article, we have studied trends in compute by curating a dataset of training compute with more than 100 milestone ML systems, and analyzing the resulting trends. Our findings suggest a more moderate rate of compute growth compared to prior work, and the emergence of a novel trend in late 2015. In particular, we identify an 18-month doubling time between 1952 and 2010, a 6-month doubling time between 2010 and 2022, and a new trend of large-scale models between late 2015 and 2022, which started 2 to 3 orders of magnitude over the previous trend and displays a 10-month doubling time. These three trends collectively span three eras of ML history: (1) the **Pre Deep Learning Era**, (2) the **Deep Learning Era**, and (3) the **Large-scale Era**.

We hope that our work will help understand the significance of different factors in driving ML progress, and encourage other researchers to use our dataset in their own analyses. In further work, we hope to continue our investigation into the inputs of ML systems Sevilla et al. [39].

Acknowledgments

We thank Alex Lyzhov, Girish Sastry, Danny Hernandez, Haydn Belfield, Jack Clark, Markus Anderljung, Alexis Carlier, Noemi Dreksler, Ben Garfinkel, Anton Korinek, Toby Shevlane, Stella Biderman, and Robert Trager. Jaime Sevilla is funded by the Open Philanthropy Project. Lennart Heim conducted part of this work during a summer fellowship in 2021 at the Stanford Existential Risks Initiative (SERI). Anson Ho conducted part of this work while funded by the Long-Term Future Fund.

References

[1] A. Chowdhery *et al.*, "Palm: Scaling language modeling with pathways," 2022.

[2] J. Hoffmann *et al.*, "Training compute-optimal large language models," 2022.

[3] J. Kaplan *et al.*, "Scaling Laws for Neural Language Models," 2020, _eprint: 2001.08361.

[4] D. Amodei and D. Hernandez, "*AI and Compute*," May 2018, published: OpenAI Blog.

[5] G. Sastry *et al.*, "*AI and Compute* Addendum: Compute Used in Older Headline Results," Nov. 2019, published: OpenAI Blog.

[6] A. Lyzhov, "*AI and Compute* Trend Isn't Predictive of What Is Happening." Apr. 2021, published: Alignment Forum (blog).

[7] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," 2020, _eprint: 2005.14165.

[8] D. Silver *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, pp. 354–359, 2017.

[9] R. Sutton, "The bitter lesson," *Incomplete Ideas (blog)*, vol. 13, p. 12, 2019.

[10] Z. Li *et al.*, "Train large, then compress: Rethinking model size for efficient training and inference of transformers," *arXiv preprint arXiv:2002.11794*, pp. 1–14, 2020.

[11] A. L. Jones, "Scaling Scaling Laws with Board Games," *arXiv preprint arXiv:2104.03113*, pp. 1–8, 2021.

[12] J. S. Rosenfeld *et al.*, "A constructive prediction of the generalization error across scales," *arXiv preprint arXiv:1909.12673*, pp. 1–30, 2019.

[13] J. Hestness *et al.*, "Deep learning scaling is predictable, empirically," *arXiv preprint arXiv:1712.00409*, pp. 1–19, 2017.

[14] J.-B. Alayrac *et al.*, "Flamingo: a visual language model for few-shot learning," 2022.

[15] N. C. Thompson *et al.*, "The Computational Limits of Deep Learning," 2020, _eprint: 2007.05558.

[16] A. Lohn and M. Musser, "How Much Longer Can Computing Power Drive Artificial Intelligence Progress?" Center for Security and Technology, Tech. Rep., Jan. 2022, published: Center for Security and Technology report https://cset.georgetown.edu/publication/ai-and-compute/.

[17] "The Akronomicon — LightOn AI Research." Akronomicon, 2022.

[18] "Computer Progress," Computer Progress, 2022.

[19] "AI Tracker," AI Tracker, 2022.

[20] R. Desislavov, F. Martínez-Plumed, and J. Hernández-Orallo, "Compute and Energy Consumption Trends in Deep Learning Inference," 2021, _eprint: 2109.05472.

[21] J. Sevilla *et al.*, "Estimating training compute of Deep Learning models," Jan. 2022.

[22] G. Moore, "\emphThe Future of Integrated Electronics," 1965, published: Electronics Magazine.

[23] M. Z. Alom *et al.*, "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches," 2018, _eprint: 1803.01164.

[24] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, Dec. 2001, pp. I–I, iSSN: 1063-6919.

[25] R. Raina, A. Madhavan, and A. Y. Ng, "Large-Scale Deep Unsupervised Learning Using Graphics Proces-

sors," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: Association for Computing Machinery, 2009, pp. 873–880, event-place: Montreal, Quebec, Canada.

[26] T. Mikolov *et al.*, "Recurrent neural network based language model," in *INTERSPEECH*, 2010.

[27] D. C. Cireşan *et al.*, "Deep, big, simple neural nets for handwritten digit recognition," *Neural Computation*, vol. 22, pp. 3207–3220, Dec. 2010.

[28] D. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification," *arXiv:1202.2745 [cs]*, Feb. 2012.

[29] D. Steinkraus, I. Buck, and P. Simard, "Using GPUs for machine learning algorithms," in *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, 2005, pp. 1115–1120 Vol. 2.

[30] K. Chellapilla, S. Puri, and P. Simard, "High Performance Convolutional Neural Networks for Document Processing," in *Tenth International Workshop on Frontiers in Handwriting Recognition*, G. Lorette, Ed. La Baule (France): Suvisoft, Oct. 2006, backup Publisher: Université de Rennes 1.

[31] D. C. Cireşan *et al.*, "Flexible, High Performance Convolutional Neural Networks for Image Classification," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, ser. IJCAI'11. AAAI Press, 2011.

[32] B. Catanzaro, N. Sundaram, and K. Keutzer, "Fast Support Vector Machine Training and Classification on Graphics Processors," in *Proceedings of the 25th international conference on Machine learning*, ser. ICML '08. New York, NY, USA: Association for Computing Machinery, 2008.

[33] L. Deng, D. Yu, and G. E. Hinton, "Deep Learning for Speech Recognition and Related Applications," Dec. 2009.

[34] G. Hinton *et al.*, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, 2012.

[35] D. H, "How Much Did AlphaGo Zero Cost," 2020.

[36] O. Vinyals *et al.*, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, pp. 350–354, Nov. 2019, number: 7782 Publisher: Nature Publishing Group.

[37] K. Wang, "DeepMind achieved StarCraft II GrandMaster Level, but at what cost?" Jan. 2020.

[38] C. Li, "OpenAI's GPT-3 Language Model: A Technical Overview," Jun. 2020.

[39] J. Sevilla, P. Villalobos, and J. F. Cerón, "\emphParameter Counts in Machine Learning," Jun. 2021, published: Alignment Forum (blog).

[40] G. Kohs *et al.*, *AlphaGo*. Moxie Pictures, Reel As Dirt, Sep. 2017.

[41] OpenAI, "AlphaStar: Grandmaster level in StarCraft II using multi-agent reinforcement learning," Oct. 2019.

[42] X. Wu, C. Zhang, and W. Du, "An Analysis on the Crisis of "Chips shortage" in Automobile Industry ——Based on the Double Influence of COVID-19 and Trade Friction," *Journal of Physics: Conference Series*, vol. 1971, Jul. 2021.

[43] K. Hazelwood *et al.*, "Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2018, pp. 620–629.

[44] D. C. Cireşan *et al.*, "Deep, Big, Simple Neural Nets for Handwritten Digit Recognition," *Neural Computation*, vol. 22, Dec. 2010.

[45] A. Ajmera and M. Ramakrishnan, "Ford to Shut Some N. American Plants for Few Weeks on Chip Shortage." Jun. 2021.

[46] K. Wiggers, "Google Trained a Trillion-Parameter AI Language Model." Jan. 2021.

[47] Y. Huang *et al.*, "GPipe: Efficient Training of Giant Neural Networks Using Pipeline Parallelism," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019, pp. 103–112, section: 1.

[48] A. Shilov, "GPU Shortages Hit Nvidia's Data Center Business: Not Enough $15,000+ GPUs." Dec. 2020, published: Tom's Hardware.

[49] O. Vinyals *et al.*, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, pp. 1–5, 2019.

[50] D. Lepikhin *et al.*, "Gshard: Scaling giant models with conditional computation and automatic sharding," 2020.

[51] "Hardware- Und Nachrichten-Links Des 30./31. Oktober 2021." 3D Center, Jan. 2022.

[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, F. Pereira *et al.*, Eds., vol. 25. Curran Associates, Inc., 2012.

[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Commun. ACM*, vol. 60, pp. 84–90, May 2017, place: New York, NY, USA Publisher: Association for Computing Machinery.

[54] D. Cireşan *et al.*, "Multi-column deep neural network for traffic sign classification," *Neural Networks*, vol. 32, pp. 333–338, 2012.

[55] "OpenAI API Pricing," OpenAI, 2021.

[56] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, 2001, pp. I–I.

[57] J. Orme, "Report: Microsoft Handed OpenAI $500m in Azure Credits." 2022, published: Techerati.

[58] C. Sun *et al.*, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 843–852.
[59] E. Barrett, "Taiwan's drought is exposing just how much water chipmakers like TSMC use (and reuse)," Jun. 2021.
[60] A. Woodie, "The Chip Shortage Seems to Be Impacting AI Workloads in the Cloud," Mar. 2021, published: Datanami.
[61] O. Sharir, B. Peleg, and Y. Shoham, "The Cost of Training NLP Models: A Concise Overview," 2020, _eprint: 2004.08900.
[62] M. G. Attinasi *et al.*, "The Semiconductor Shortage and Its Implication for Euro Area Trade, Production and Prices." Jun. 2021, published: ECB Economic Bulletin.
[63] S. Athlur *et al.*, "Varuna: Scalable, Low-cost Training of Massive Deep Learning Models," 2021, _eprint: 2111.04007.
[64] N. Patel, "Why the Global Chip Shortage is Making It So Hard to Buy a PS5," Aug. 2021.
[65] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, pp. 386–408, 1958, place: US Publisher: American Psychological Association.
[66] D. Klein, "Mighty mouse."
[67] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, vol. 3, pp. 210–229, Jul. 1959, conference Name: IBM Journal of Research and Development.
[68] O. G. Selfridge, "Pandemonium: A Paradigm for Learning | AITopics."
[69] B. Widrow and M. E. Hoff, "Adaptive switching circuits," in *Neurocomputing: foundations of research*. Cambridge, MA, USA: MIT Press, Jan. 1988, pp. 123–134.
[70] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, pp. 193–202, Apr. 1980.
[71] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986, number: 6088 Publisher: Nature Publishing Group.
[72] T. J, Sejnowski, and C. R. Rosenberg, "Parallel Networks that Learn to Pronounce English Text."
[73] Y. LeCun *et al.*, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, pp. 541–551, Dec. 1989, conference Name: Neural Computation.
[74] D. A. Pomerleau, "ALVINN: An Autonomous Land Vehicle in a Neural Network," in *Advances in Neural Information Processing Systems*, vol. 1. Morgan-Kaufmann, 1988.
[75] G. Tesauro, "Practical issues in temporal difference learning," *Machine Learning*, vol. 8, pp. 257–277, May 1992.
[76] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 23–38, Jan. 1998, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.
[77] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, pp. 1735–1780, Nov. 1997.
[78] S.-H. Chen, S.-H. Hwang, and Y.-R. Wang, "An RNN-based prosodic information synthesizer for Mandarin text-to-speech," *IEEE Transactions on Speech and Audio Processing*, vol. 6, pp. 226–239, May 1998, conference Name: IEEE Transactions on Speech and Audio Processing.
[79] Y. Lecun *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, Nov. 1998, conference Name: Proceedings of the IEEE.
[80] Y. Bengio *et al.*, "A neural probabilistic language model," *The Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, Mar. 2003.
[81] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, pp. 602–610, Jul. 2005.
[82]
[83] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, Mar. 2010, pp. 249–256, iSSN: 1938-7228.
[84] G. E. Hinton *et al.*, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv:1207.0580 [cs]*, Jul. 2012.
[85]
[86] V. Mnih *et al.*, "Playing Atari with Deep Reinforcement Learning," *arXiv:1312.5602 [cs]*, Dec. 2013.
[87] T. Mikolov *et al.*, "Distributed Representations of Words and Phrases and their Compositionality," in *NIPS*, 2013.
[88] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," *arXiv:1311.2901 [cs]*, Nov. 2013.
[89] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv:1312.6114 [cs, stat]*, May 2014.
[90] D. C. Cireşan *et al.*, "Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*, ser. Lecture Notes in Computer Science, K. Mori *et al.*, Eds. Berlin, Heidelberg: Springer, 2013, pp. 411–418.
[91] I. J. Goodfellow *et al.*, "Generative Adversarial Net-

works," *arXiv:1406.2661 [cs, stat]*, Jun. 2014.
[92] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *arXiv:1409.0473 [cs, stat]*, May 2016.
[93] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Apr. 2015.
[94] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," *arXiv:1409.3215 [cs]*, Dec. 2014.
[95] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Jan. 2017.
[96] K. He *et al.*, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *arXiv:1502.01852 [cs]*, Feb. 2015.
[97] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9, iSSN: 1063-6919.
[98] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *arXiv:1509.02971 [cs, stat]*, Jul. 2019.
[99] D. Amodei *et al.*, "Deep Speech 2: End-to-End Speech Recognition in English and Mandarin," *arXiv:1512.02595 [cs]*, Dec. 2015.
[100] K. He *et al.*, "Deep Residual Learning for Image Recognition," *arXiv:1512.03385 [cs]*, Dec. 2015.
[101] K. He *et al.*, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *arXiv:1406.4729 [cs]*, vol. 8691, pp. 346–361, 2014.
[102] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, Jan. 2016, number: 7587 Publisher: Nature Publishing Group.
[103] Y. Wu *et al.*, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," *CoRR*, vol. abs/1609.08144, 2016.
[104] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *arXiv:1610.02357 [cs]*, Apr. 2017.
[105] B. Zoph and Q. V. Le, "Neural Architecture Search with Reinforcement Learning," *arXiv:1611.01578 [cs]*, Feb. 2017.
[106] N. Brown and T. Sandholm, "Libratus: The Superhuman AI for No-Limit Poker," pp. 5226–5228, 2017.
[107]
[108] M. Moravčík *et al.*, "DeepStack: Expert-Level Artificial Intelligence in No-Limit Poker," *Science*, vol. 356, pp. 508–513, May 2017.
[109] N. Shazeer *et al.*, "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer," *arXiv:1701.06538 [cs, stat]*, Jan. 2017.
[110] A. Vaswani *et al.*, "Attention Is All You Need," *arXiv:1706.03762 [cs]*, Dec. 2017.
[111] C. Sun *et al.*, "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era," *arXiv:1707.02968 [cs]*, Aug. 2017.
[112]
[113]
[114] L. Espeholt *et al.*, "IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures," *arXiv:1802.01561 [cs]*, Jun. 2018.
[115] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv:1804.02767 [cs]*, Apr. 2018.
[116] A. Radford, "Improving Language Understanding with Unsupervised Learning," Jun. 2018.
[117] M. Jaderberg *et al.*, "Human-level performance in first-person multiplayer games with population-based deep reinforcement learning," *Science*, vol. 364, pp. 859–865, May 2019.
[118] A. Brock, J. Donahue, and K. Simonyan, "Large Scale GAN Training for High Fidelity Natural Image Synthesis," *arXiv:1809.11096 [cs, stat]*, Feb. 2019.
[119] J. Devlin *et al.*, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Oct. 2018.
[120] C. Liu *et al.*, "Progressive Neural Architecture Search," Dec. 2017.
[121] E. Real *et al.*, "Regularized Evolution for Image Classifier Architecture Search," Feb. 2018.
[122] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," Nov. 2017.
[123] N. Bard *et al.*, "The Hanabi Challenge: A New Frontier for AI Research," Feb. 2019.
[124] A. Radford *et al.*, "Better Language Models and Their Implications," Feb. 2019.
[125] H. Cai, L. Zhu, and S. Han, "ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware," Dec. 2018.
[126]
[127] M. Naumov *et al.*, "Deep Learning Recommendation Model for Personalization and Recommendation Systems," May 2019.
[128] A. Barbu *et al.*, "ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.
[129] B. Baker *et al.*, "Emergent Tool Use from Multi-Agent Interaction," Sep. 2019.
[130] M. Shoeybi *et al.*, "Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism," Sep. 2019.
[131] L. Wang *et al.*, "AlphaX: eXploring Neural Architectures with Deep Neural Networks and Monte Carlo Tree Search," Mar. 2019.
[132] OpenAI *et al.*, "Solving Rubik's Cube with a Robot Hand."
[133] C. Raffel *et al.*, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," Oct. 2019.
[134] OpenAI *et al.*, "Dota 2 with Large Scale Deep Rein-

forcement Learning," Dec. 2019.
[135] A. W. Senior *et al.*, "Improved protein structure prediction using potentials from deep learning," *Nature*, vol. 577, pp. 706–710, Jan. 2020, number: 7792 Publisher: Nature Publishing Group.
[136] D. Adiwardana and T. Luong, "Towards a Conversational Agent that Can Chat About…Anything," 2020.
[137] C. Rosset, "Turing-NLG: A 17-billion-parameter language model by Microsoft," Feb. 2020.
[138] A. Madani *et al.*, "ProGen: Language Modeling for Protein Generation," bioRxiv, Tech. Rep., Mar. 2020, section: New Results Type: article.
[139] H. Cai *et al.*, "Once-for-All: Train One Network and Specialize it for Efficient Deployment," Aug. 2019.
[140] M. Chen, A. Radford, and I. Sutskever, "Image GPT," Jun. 2020.
[141]
[142] A. Baevski *et al.*, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," Jun. 2020.
[143] Z. Zhang *et al.*, "CPM: A Large-scale Generative Chinese Pre-trained Language Model," Dec. 2020.
[144] W. Antoun, F. Baly, and H. Hajj, "AraGPT2: Pre-Trained Transformer for Arabic Language Generation," Dec. 2020.
[145] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," Sep. 2020.
[146] H. Pham *et al.*, "Meta Pseudo Labels," Mar. 2020.
[147] D. Mudigere *et al.*, "Software-Hardware Co-design for Fast and Scalable Training of Deep Learning Recommendation Models," Apr. 2021.
[148] D. R. So *et al.*, "Primer: Searching for Efficient Transformers for Language Modeling," Sep. 2021.
[149] A. Radford *et al.*, "Learning Transferable Visual Models From Natural Language Supervision," Feb. 2021.
[150] A. R. nad Mikhail Pavlov, G. Goh, and S. Gray, "DALL·E: Creating Images from Text."
[151] W. Fedus, B. Zoph, and N. Shazeer, "Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity," Jan. 2021.
[152] "GPT-Neo," Eleuther AI.
[153] W. Zeng *et al.*, "PanGu-α: Large-scale Autoregressive Pretrained Chinese Language Models with Auto-parallel Computation," Apr. 2021.
[154] A. Komatsuzaki, "GPT-J-6B: 6B JAX-Based Transformer," Jun. 2021.
[155] "Naver Corporation," Naver Corporation.
[156] M. Ding *et al.*, "CogView: Mastering Text-to-Image Generation via Transformers," May 2021.
[157] X. Zhai *et al.*, "Scaling Vision Transformers," Jun. 2021.
[158] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," Jun. 2020.
[159] "Baidu Research," Baidu Research.
[160] W.-N. Hsu *et al.*, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units," Jun. 2021.
[161] P. Goyal *et al.*, "Self-supervised Pretraining of Visual Features in the Wild," Mar. 2021.
[162] O. Lieber *et al.*, "Announcing AI21 Studio and Jurassic-1 Language Models."
[163] J. Lin *et al.*, "M6-10T: A Sharing-Delinking Paradigm for Efficient Multi-Trillion Parameter Pretraining," Oct. 2021.
[164] A. Alvi and P. Kharya, "Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, the World's Largest and Most Powerful Generative Language Model," Oct. 2021.
[165] S. Wu *et al.*, "Yuan 1.0: Large-Scale Pre-trained Language Model in Zero-Shot and Few-Shot Learning," Oct. 2021.
[166] J. Rae, G. Irving, and L. Weidinger, "Language modelling at scale."
[167] Z. Zhang *et al.*, "Aggregating Nested Transformers," May 2021.
[168] C. Jia *et al.*, "Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision," Feb. 2021.
[169] Z. Lan *et al.*, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," Sep. 2019.
[170] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," Jul. 2016.
[171] T. Schuster *et al.*, "Cross-Lingual Alignment of Contextual Word Embeddings, with Applications to Zero-shot Dependency Parsing," Feb. 2019.
[172] J. Dai *et al.*, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," May 2016.
[173] X. Wang *et al.*, "KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation," Nov. 2019.
[174] A. Bordes *et al.*, "Translating Embeddings for Modeling Multi-relational Data," in *Advances in Neural Information Processing Systems*, vol. 26. Curran Associates, Inc., 2013.
[175] R. Thoppilan *et al.*, "LaMDA: Language Models for Dialog Applications," Jan. 2022.
[176] C. Leahy, "Announcing GPT-NeoX-20B," Feb. 2022.