COMP 690-204 Deep Generative Models

Lecture 3: AR Reading

Jason Ren

Fall 2026, CS, UNC

The University
of North Carolina
at Chapel Hill

Training language models to follow instructions with human feedback

L Ouyang, J Wu, X Jiang, D Almeida, C Wainwright, P Mishkin, C Zhang, S Agarwal…

Advances in neural information processing systems, 2022 · proceedings.neurips.cc

Abstract

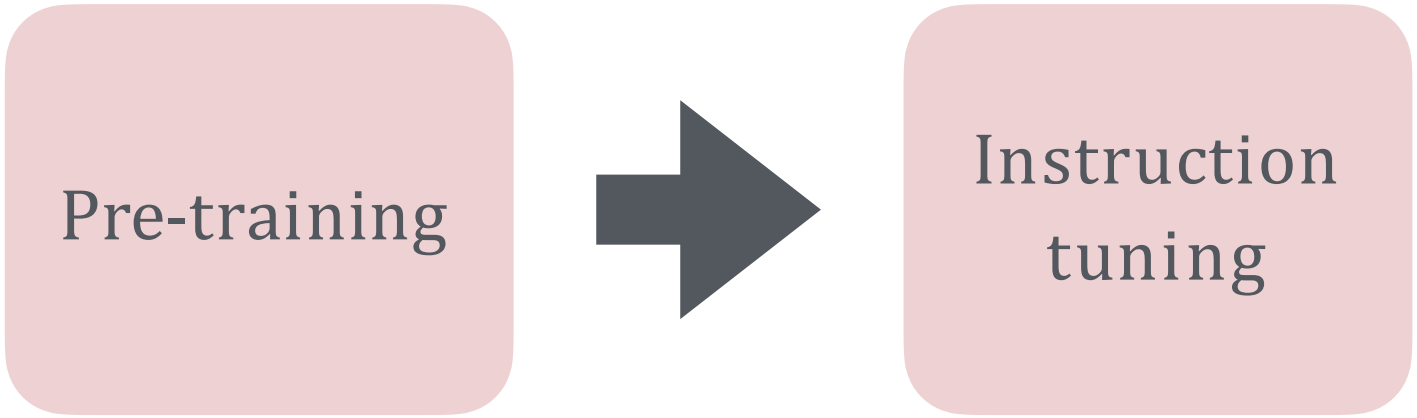
Making language models bigger does not inherently make them better at following a user's intent. For example, large language models can generate outputs that are untruthful, toxic, or simply not helpful to the user. In other words, these models are not aligned with their users. In this paper, we show an avenue for aligning language models with user intent on a wide range of tasks by fine-tuning with human feedback. Starting with a set of labeler-written prompts and prompts submitted through a language model API, we collect a

SHOW MORE

Save Cite Cited by 30819 Related articles All 29 versions

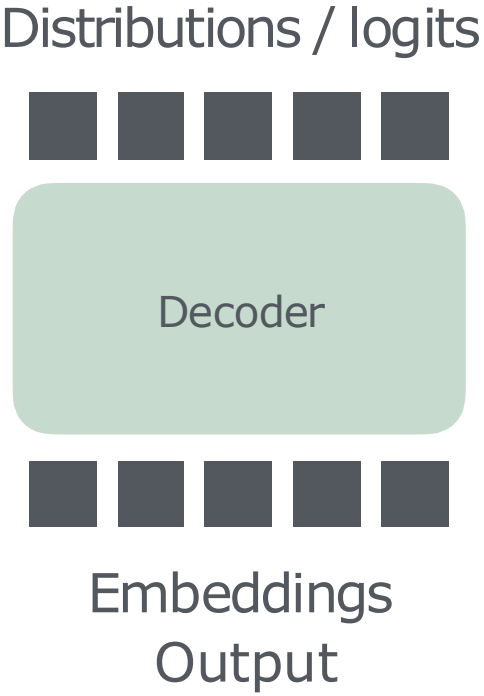
*Acknowledgement: Many of the following slides are modified from Philipp Krähenbühl (UT Austin)

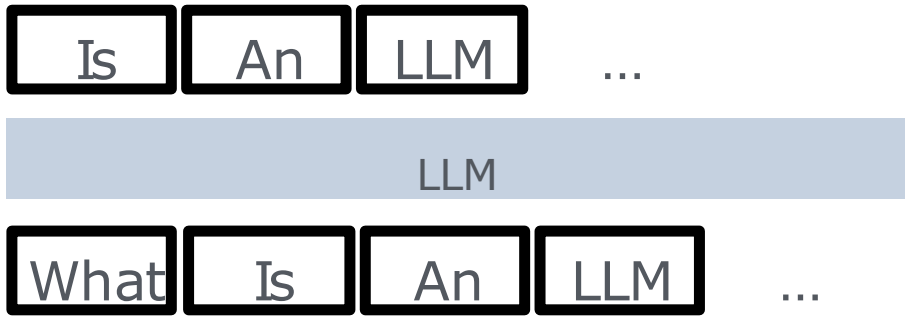
Instruction tuning



Language Models

- Decoder-only LLMs
  - Modeling auto-regressive distribution over tokens
  - $P(\mathbf{t}) = P(t_1)P(t_2|t_1)P(t_3|t_1,t_2)P(t_4|t_1 \ldots t_3) \ldots$
- Generation / Sampling: $\mathbf{t} \sim P$





Instruction tuning

- Teach a LLM to follow a certain template in answering
  - Tell LLM **what to do**
  - Does **not** tell LLM **what not to do**

> I want to refocus my diet on only eating Apples and supplements.

```
Let me provide information on how to make the most of eating apples as a primary food source.

**Apple selection:** Focus on eating a variety of apples to ensure you get a range of nutrients. Some apple varieties are higher in certain nutrients than others. For example:

* Granny Smith and Braeburn apples have more malic acid, which can help with digestion.

* Gala and Fuji apples contain more vitamins A and C.

…
```

RLHF

Reinforcement Learning from Human Feedback

- Shape LLM outputs according to human preference / ranking

RLHF

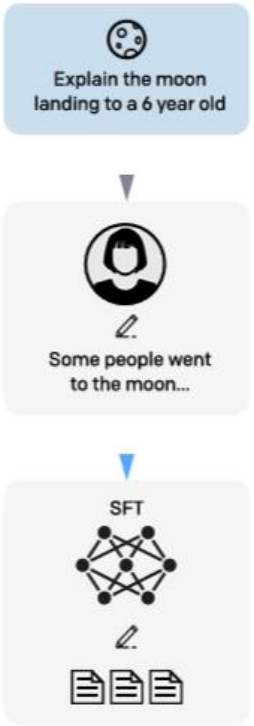
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.



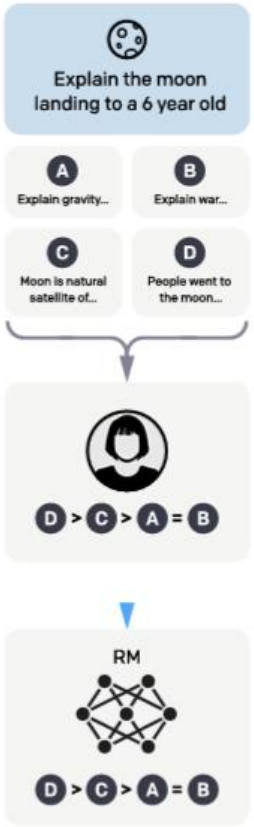
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

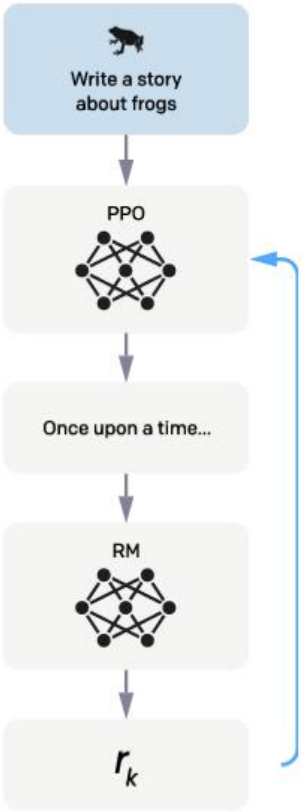
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

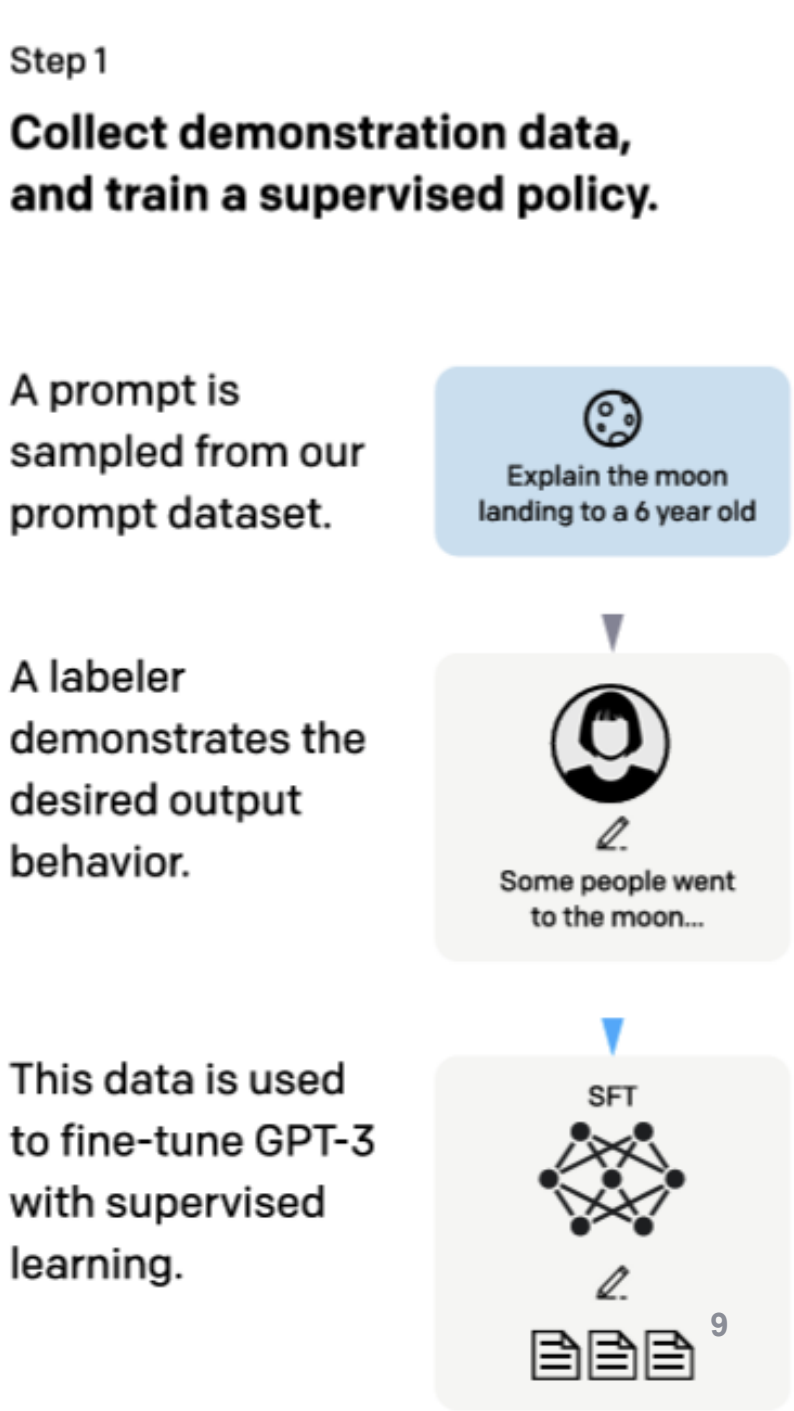
The reward is used to update the policy using PPO.



RLHF

Reinforcement Learning from Human Feedback

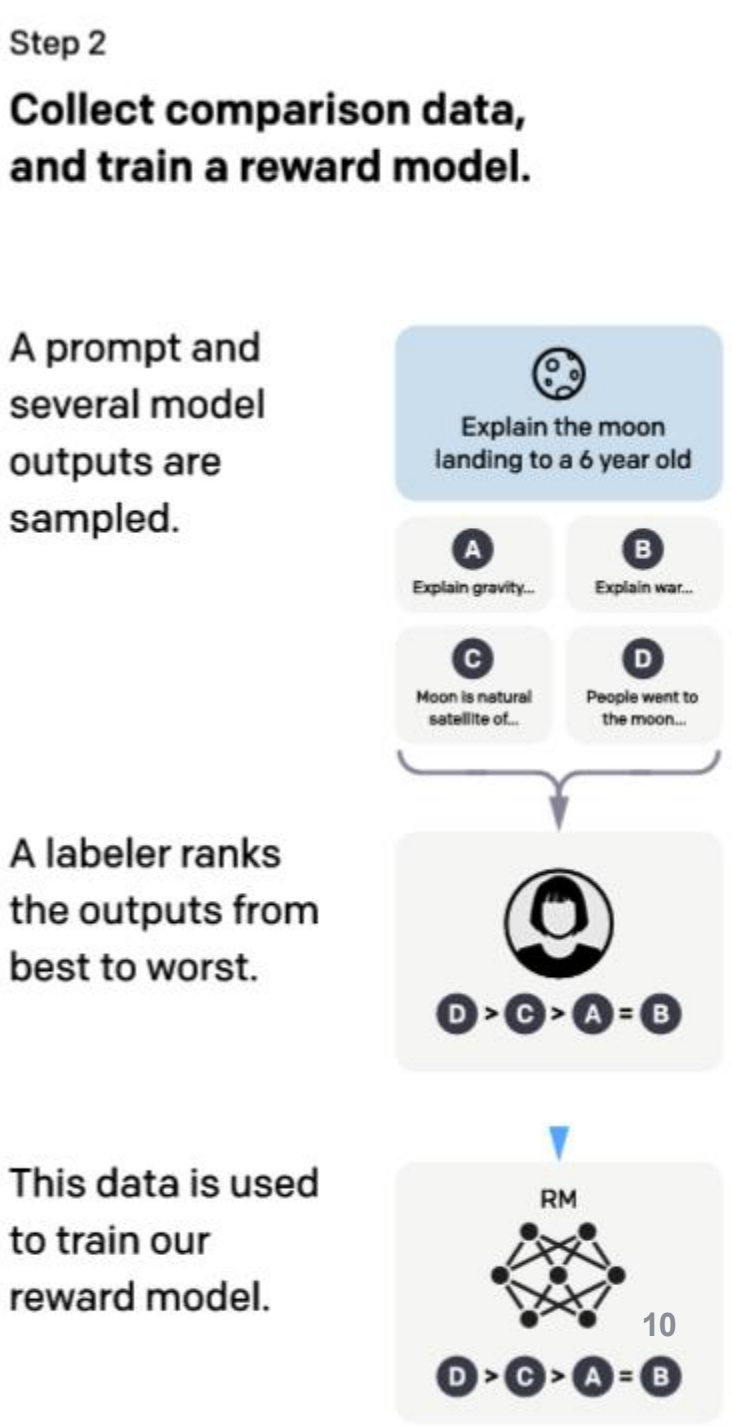
- Step 1: Instruction tuning
  - Human labeler writes prompt
    - Plain, few-shot, customer-based
  - Human labeler writes answer
  - InstructGPT: 13k samples



RLHF

Reinforcement Learning from Human Feedback

- Step 2: Reward model learning
  - Human labeler writes prompt
    - Plain, few-shot, customer-based
  - Human labeler ranks ansers
  - InstructGPT: 33k samples (6.6k annotator, 26.5k customer)

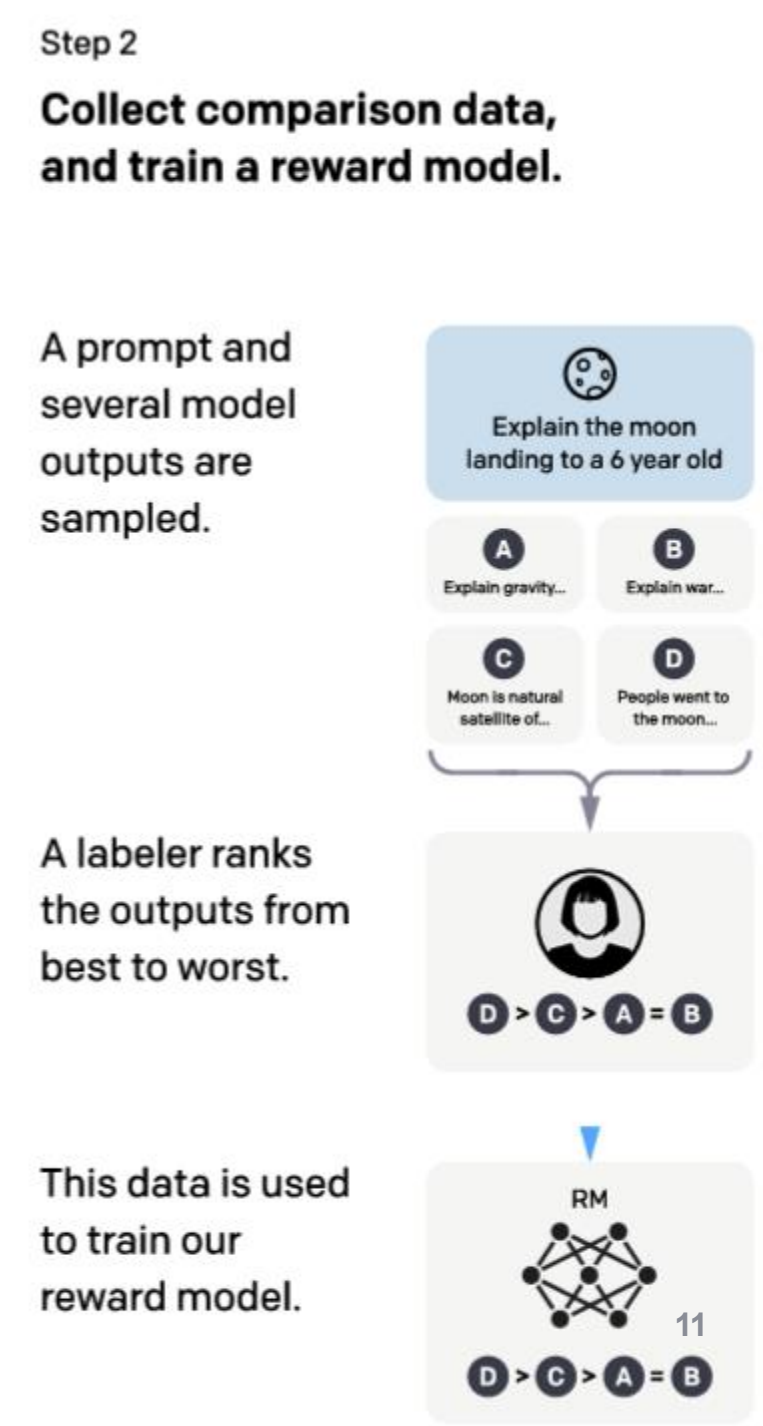


RLHF

Reinforcement Learning from Human Feedback

- Step 2: Reward model learning
  - Train a small 6B reward model $r(x, y)$
    - LLM is 175B
  - Loss pairwise preference (Bradley-Terry model)

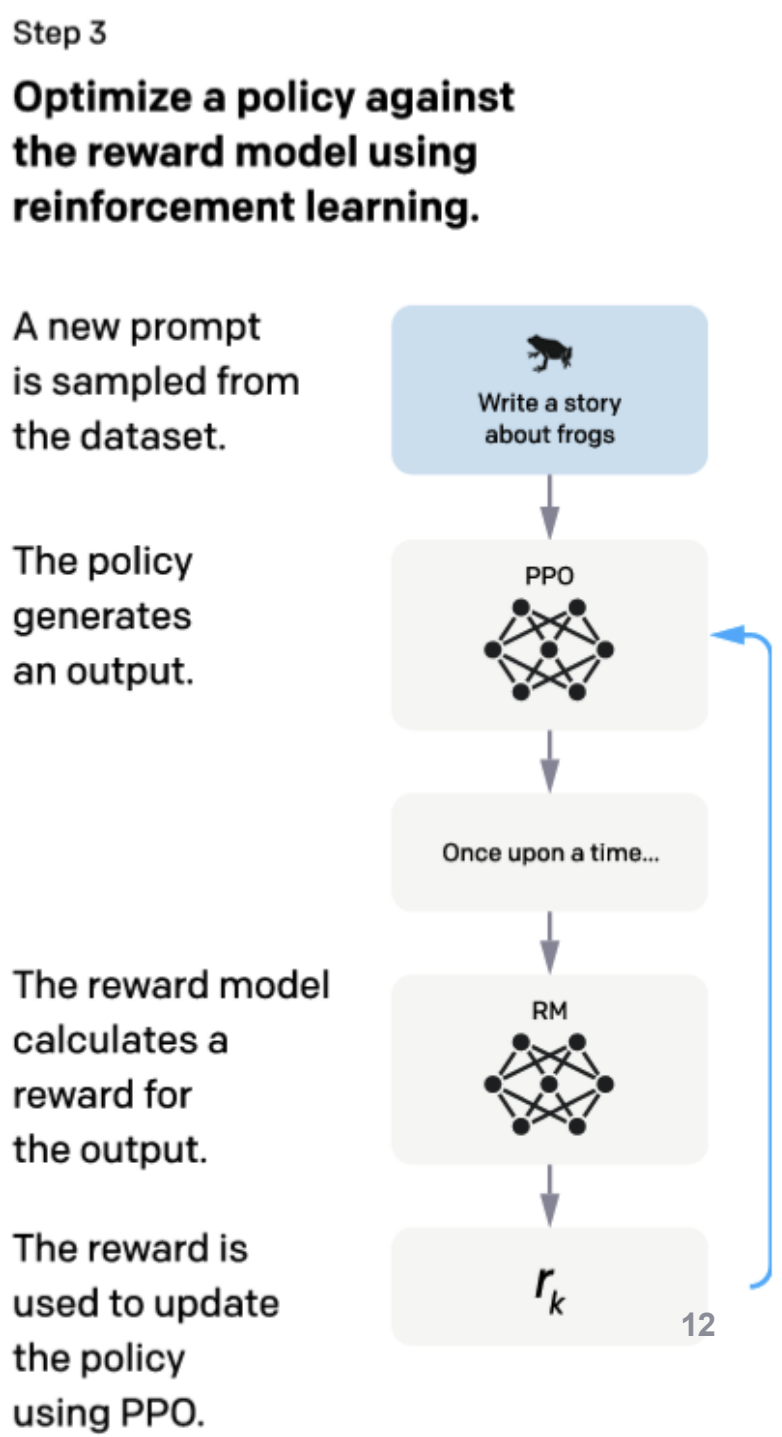
$$\ell = E_{x,y_+,y_-}\left[\log \sigma\left(r(x, y_+) - r(x, y_-)\right)\right]$$



RLHF

Reinforcement Learning from Human Feedback

- Step 3: Reinforcement Learning
  - Collect interesting prompts
  - InstructGPT: 32k samples (customer data)



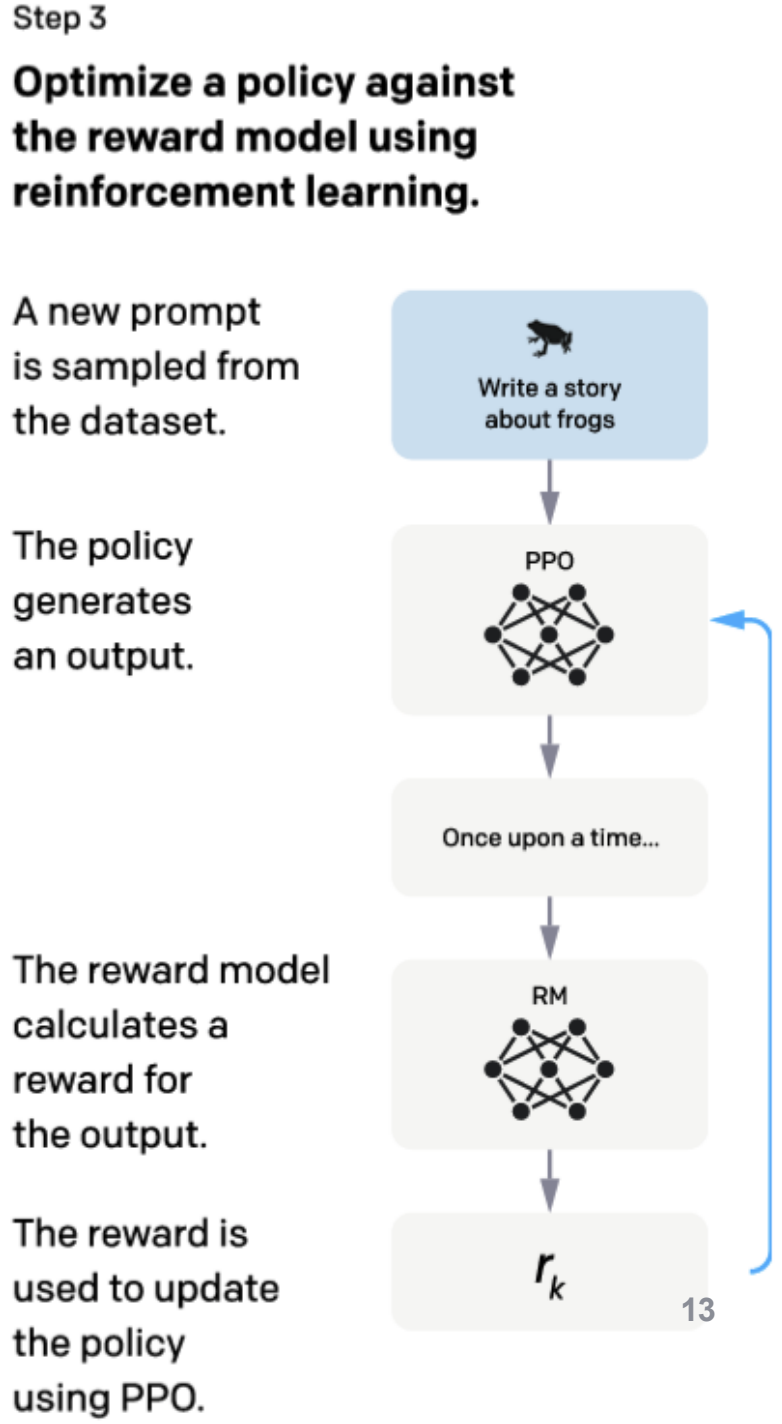
RLHF

Reinforcement Learning from Human Feedback

- Step 3: Reinforcement Learning
  - Fine-tune LLM to maximize reward model $r(x, y)$
  - PPO maximize:

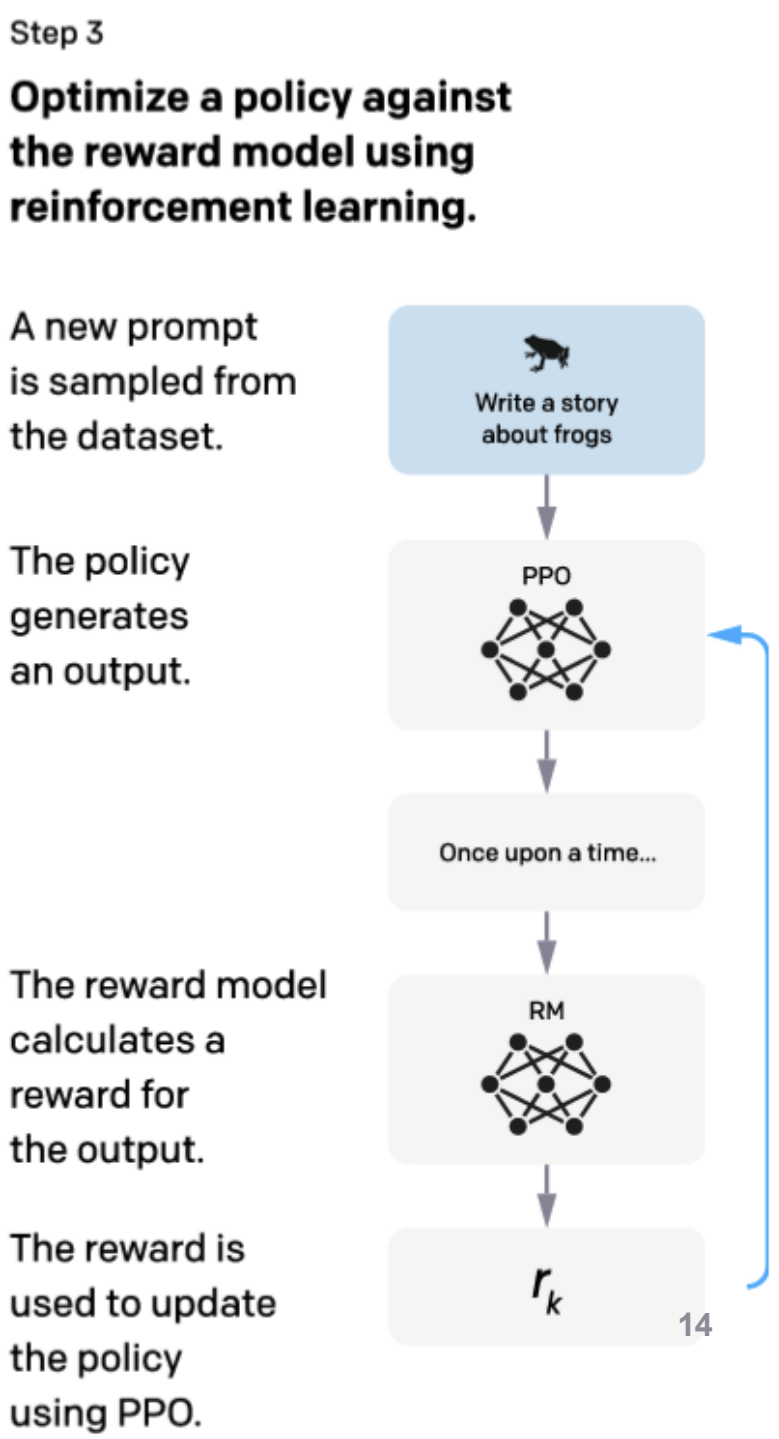
$$E_{y\sim P(\cdot|x)}\left[(r(y,x))\nabla \log P(y|x)\right] - \beta D_{KL}\left[P(y|x)|P_{ref}(y|x)\right]$$

    - Action =predict next token
    - Requires 4 models: Reference, generator, critic, reward



Why use reinforcement learning?

- Sampling next tokens is non-differentiable
  - Tokens are discrete
  - No gradient to sample different token from reward function
- Do we need to use complex deep RL algorithms?

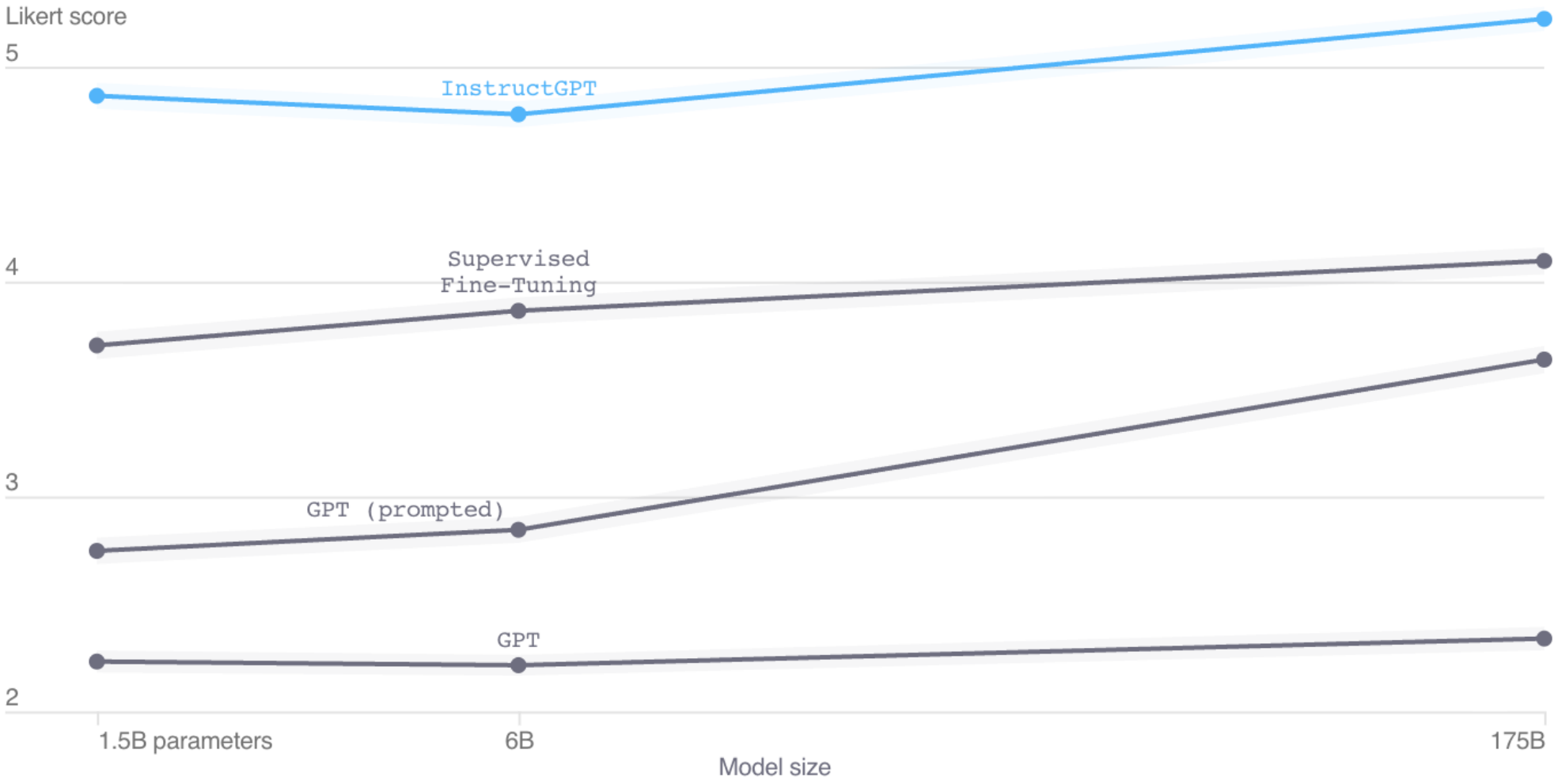


RLHF

Reinforcement Learning from Human Feedback

- RLHF alone degrades models performance: Alignment Tax
  - aligning the models only on customer tasks can make their performance worse on some other academic NLP tasks
- Solution:
  - Add KL-divergence penalty between Instruction-tuned and RLHF model
  - Mix in pre-training data / gradient

RLHF


Likert score
5
4
3
2
InstructGPT
Supervised
Fine-Tuning
GPT (prompted)
GPT
1.5B parameters
6B
175B
Model size

RLHF

Dataset

RealToxicity

GPT	0.233
Supervised Fine-Tuning	0.199
InstructGPT	**0.196**

Dataset

TruthfulQA

GPT	0.224
Supervised Fine-Tuning	0.206
InstructGPT	**0.413**

API Dataset

Hallucinations

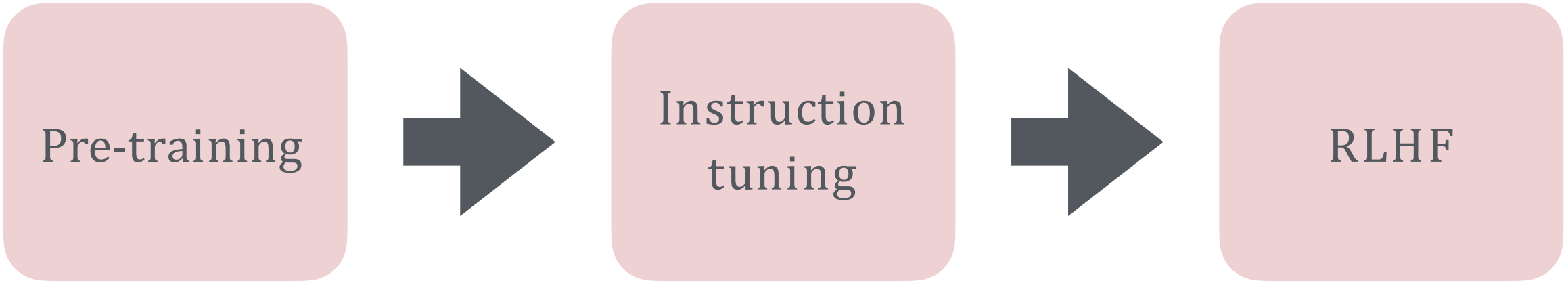
GPT	0.414
Supervised Fine-Tuning	**0.078**
InstructGPT	0.172

API Dataset

Customer Assistant Appropriate

GPT	0.811
Supervised Fine-Tuning	0.880
InstructGPT	**0.902**

RLHF



Training compute-optimal large language models

J Hoffmann, S Borgeaud, A Mensch, E Buchatskaya, T Cai, E Rutherford, DL Casas…

arXiv preprint arXiv:2203.15556, 2022 · arxiv.org

> We investigate the optimal model size and number of tokens for training a transformer language model under a given compute budget. We find that current large language models are significantly undertrained, a consequence of the recent focus on scaling language models whilst keeping the amount of training data constant. By training over 400 language models ranging from 70 million to over 16 billion parameters on 5 to 500 billion tokens, we find that for compute-optimal training, the model size and the number of

SHOW MORE

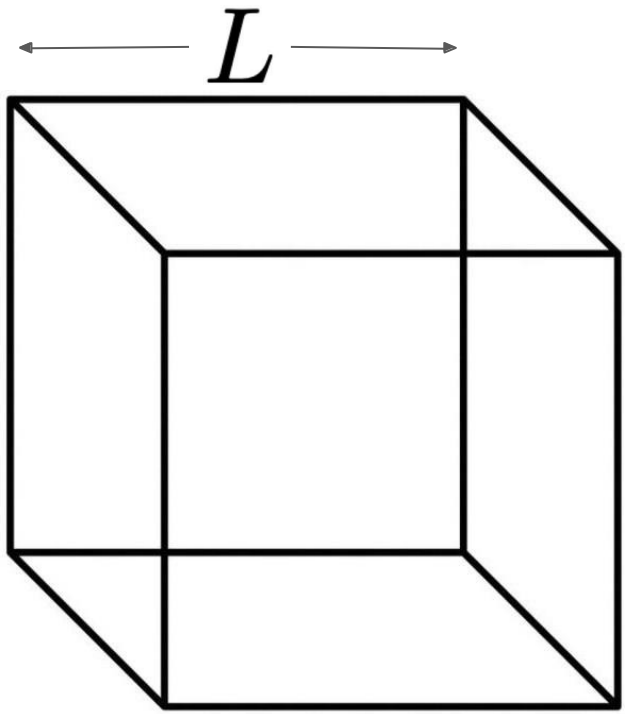
☆ Save Cite Cited by 5315 Related articles All 15 versions »

*Acknowledgement: Many of the following slides are modified from Sourav Biswas and Ben Agro

What are Scaling Laws ?

- Describe how system properties change w.r.t. parameters
- Commonly used across various fields
  - Physics
  - Statistics
  - Biology

$$A \propto L^2$$
$$M \propto L^3$$



Prior Works on Scaling Laws in Neural Networks

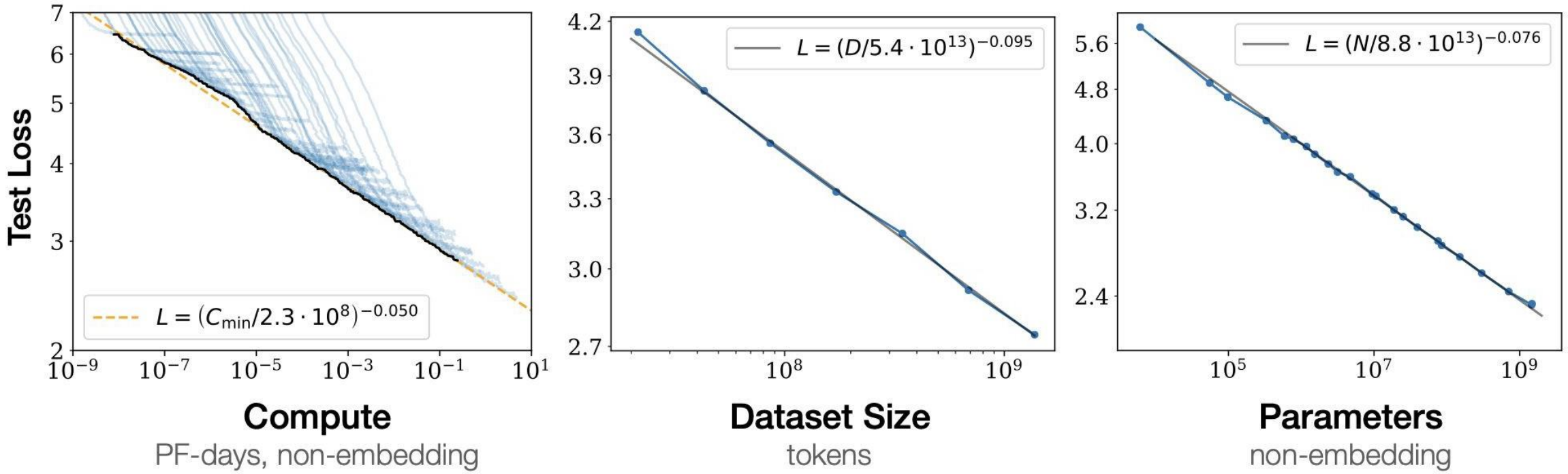
- Neural networks (1989)
- Generalization of neural nets (1991)
- Classification (1993)
- General deep learning (2017)
- Generative modeling (2020)
- Language models (2020)

$$L \propto f(N)$$
$$L \propto g(D)$$

Ahmad et al. "Scaling and Generalization in Neural Networks: A Case Study." (1989).
Cohn et al. "Can neural networks do better than the Vapnik-Chervonenkis bounds." (1991).
Barkai et al. "Scaling Laws in Learning of Classification Tasks." (1993).
Hestness et al. "Deep Learning Scaling is Predictable, Empirically." (2017).
Henighan et al. "Scaling Laws for Autoregressive Generative Modeling." (2020).
Kaplan et al. "Scaling Laws for Neural Language Models." (2020).

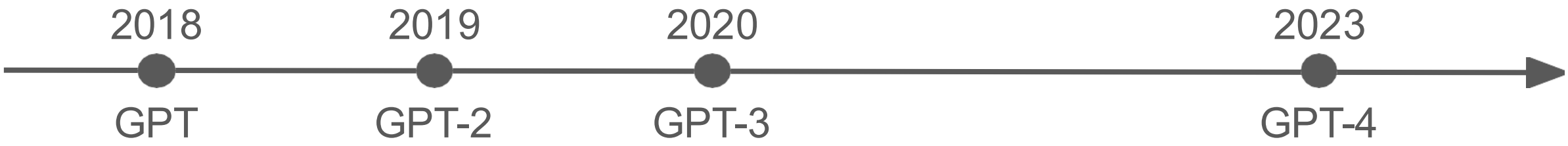
Existing Work on Scaling Laws

- Kaplan et al. show a relationship between model size & performance
  - This motivated much of the progress and scaling of LLMs



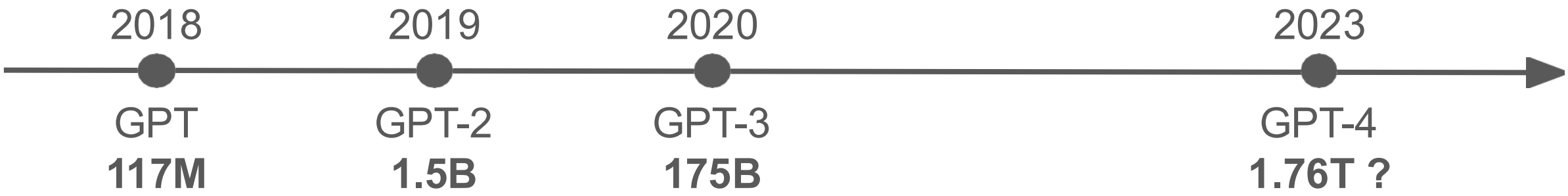
Kaplan et al. "Scaling Laws for Neural Language Models." *arXiv preprint arXiv:2001.08361* (2020).

History of LLM Progression



Layton, D. "ChatGPT — How we got to where we are today — a timeline of GPT development."
https://medium.com/@dlaytonj2/chatgpt-how-we-got-to-where-we-are-today-a-timeline-of-gpt-development-f7a35dcc660e (2023).
Lubbad, M. "GPT-4 Parameters: Unlimited guide NLP's Game-Changer."
https://mlubbad.medium.com/the-ultimate-guide-to-gpt-4-parameters-everything-you-need-to-know-about-nlps-game-changer-109b8767855a (2023).
Shree, P. "The Journey of Open AI GPT models." https://medium.com/walmartglobaltech/the-journey-of-open-ai-gpt-models-32d95b7b7fb2 (2020).

History of LLM Progression



Layton, D. "ChatGPT — How we got to where we are today — a timeline of GPT development." https://medium.com/@dlaytonj2/chatgpt-how-we-got-to-where-we-are-today-a-timeline-of-gpt-development-f7a35dcc660e (2023).
Lubbad, M. "GPT-4 Parameters: Unlimited guide NLP's Game-Changer." https://mlubbad.medium.com/the-ultimate-guide-to-gpt-4-parameters-everything-you-need-to-know-about-nlps-game-changer-109b8767855a (2023).
Shree, P. "The Journey of Open AI GPT models." https://medium.com/walmartglobaltech/the-journey-of-open-ai-gpt-models-32d95b7b7fb2 (2020).

Trends

- Larger models & more data
- Hyperparameter tuning
- Different architectures

Model	Size (# Parameters)	Training Tokens
LaMDA (Thoppilan et al., 2022)	137 Billion	168 Billion
GPT-3 (Brown et al., 2020)	175 Billion	300 Billion
Jurassic (Lieber et al., 2021)	178 Billion	300 Billion
Gopher (Rae et al., 2021)	280 Billion	300 Billion
MT-NLG 530B (Smith et al., 2022)	530 Billion	270 Billion
Chinchilla	70 Billion	1.4 Trillion

Yang et al. "Tuning large neural networks via zero-shot hyperparameter transfer." (2021).
McCandlish et al. "An empirical model of large-batch training." (2018).
Shallue et al. "Measuring the effects of data parallelism on neural network training." (2018).
Zhang et al. "Which algorithmic choices matter at which batch sizes?." (2019).
Levine et al. "The depth-to-width interplay in self-attention." (2020).

Key Idea

- Motivation: LLM training is expensive
  - Large models, with parameter count **N**
  - Large datasets, with **D** tokens
  - For a given compute budget **C** (in FLOPs)

$$N_{opt}(C), D_{opt}(C) = \underset{N,D \text{ s.t. } \mathrm{FLOPs}(N,D)=C}{\operatorname{argmin}} L(N, D)$$

Differences from Previous Works

This paper:

- **Varies** number of training tokens
- **Adapts** cosine LR schedule
- **Larger** models
- **Considers** embedding parameters

Kaplan et al:

- **Fixed** number of training tokens
- **Fixed** cosine LR schedule
- **Smaller** models
- **Doesn't count** embedding parameters

Kaplan et al. "Scaling Laws for Neural Language Models." *arXiv preprint arXiv:2001.08361* (2020).

Key Idea

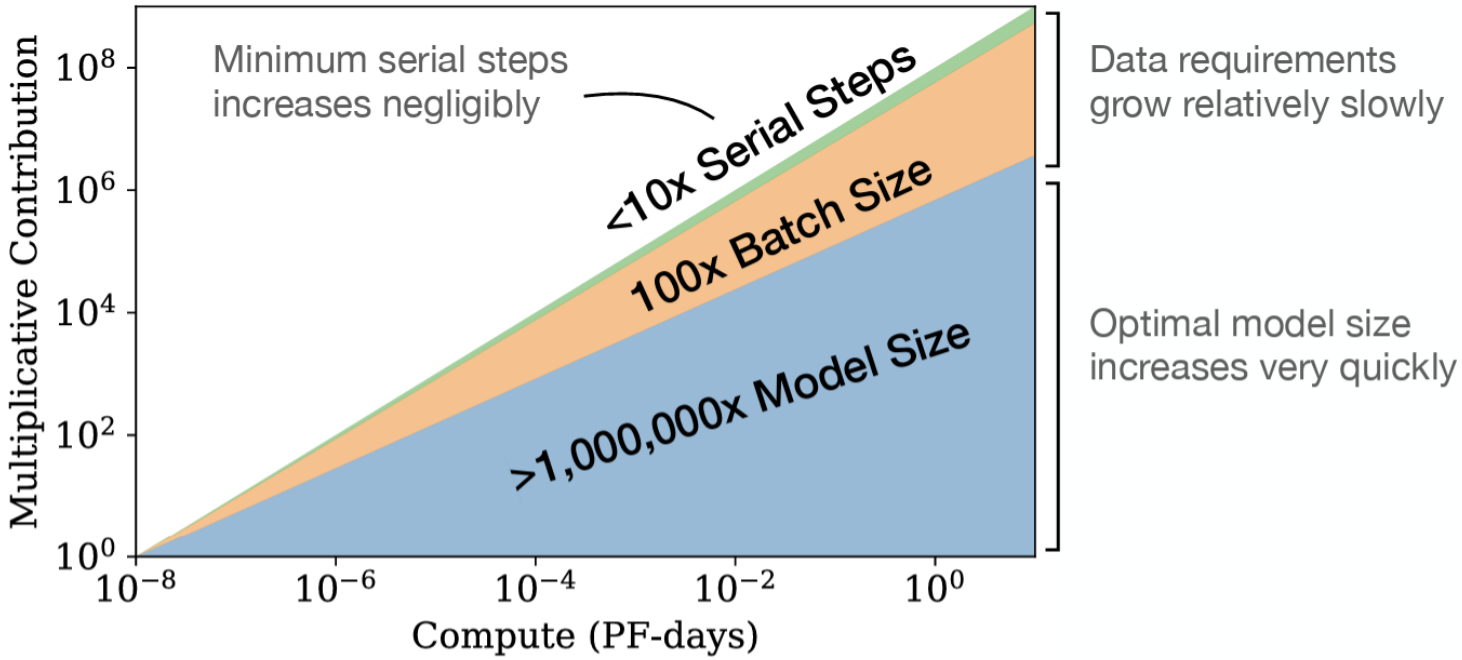
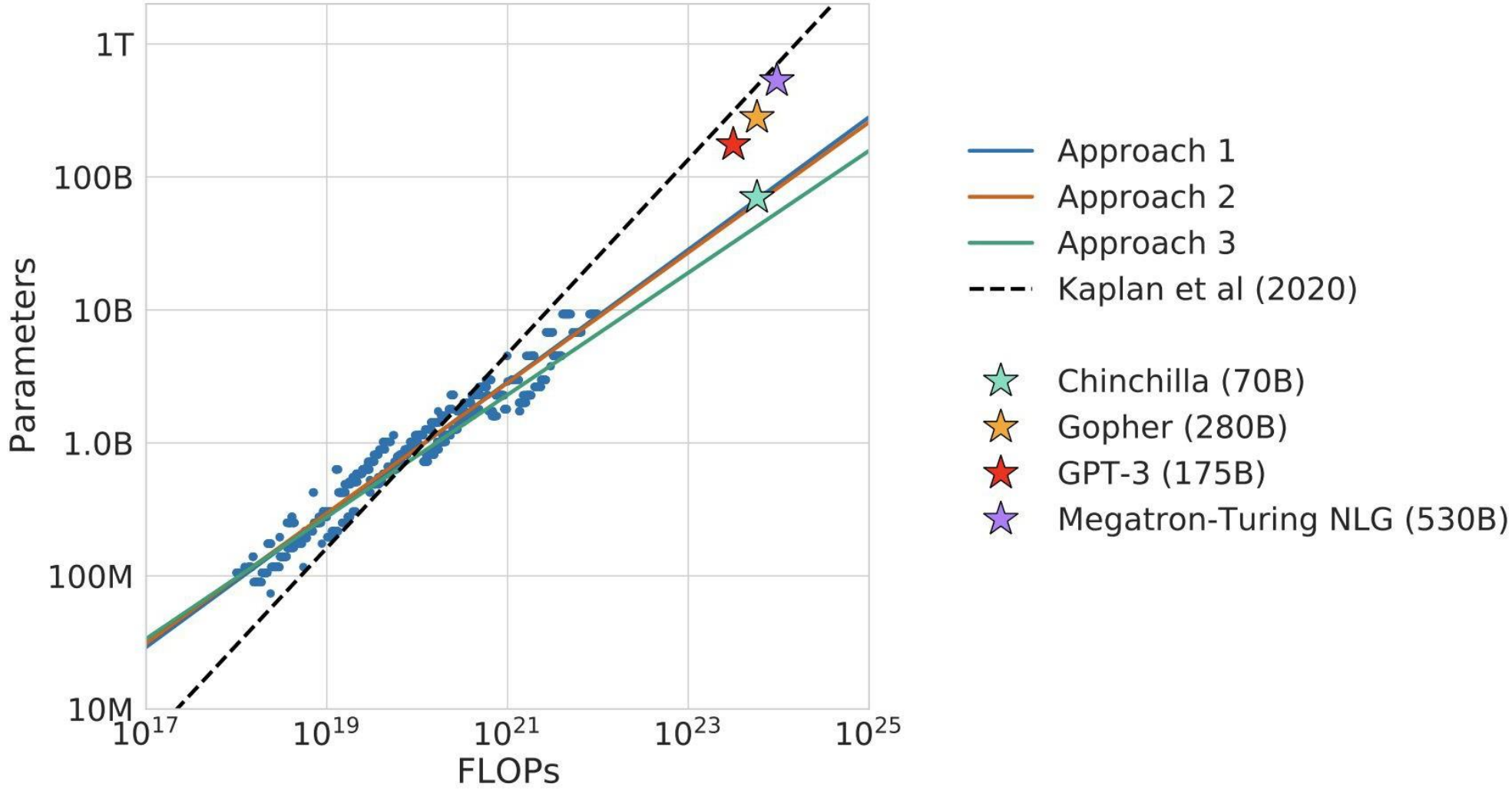


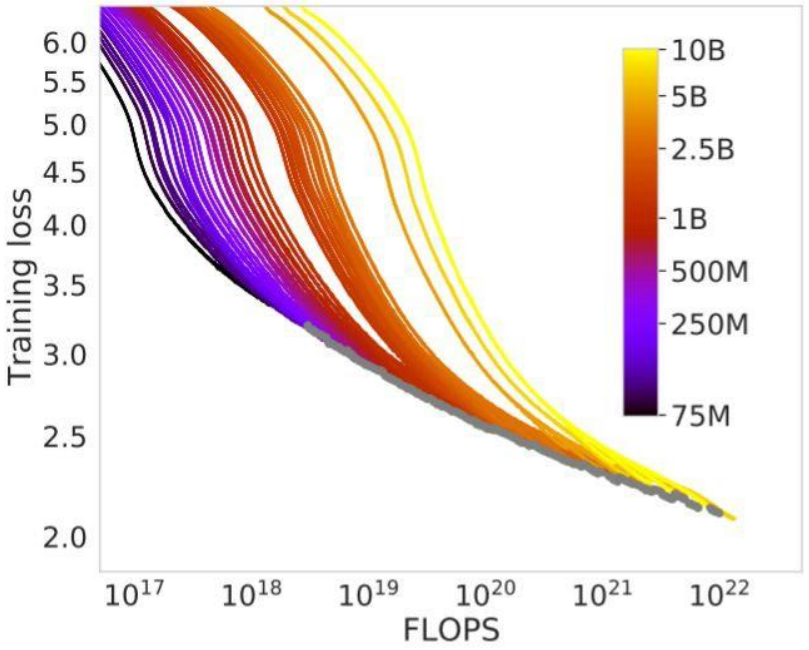
Figure 3 As more compute becomes available, we can choose how much to allocate towards training larger models, using larger batches, and training for more steps. We illustrate this for a billion-fold increase in compute. For optimally compute-efficient training, most of the increase should go towards increased model size. A relatively small increase in data is needed to avoid reuse. Of the increase in data, most can be used to increase parallelism through larger batch sizes, with only a very small increase in serial training time required.

Kaplan et al. "Scaling Laws for Neural Language Models." *arXiv preprint arXiv:2001.08361* (2020).

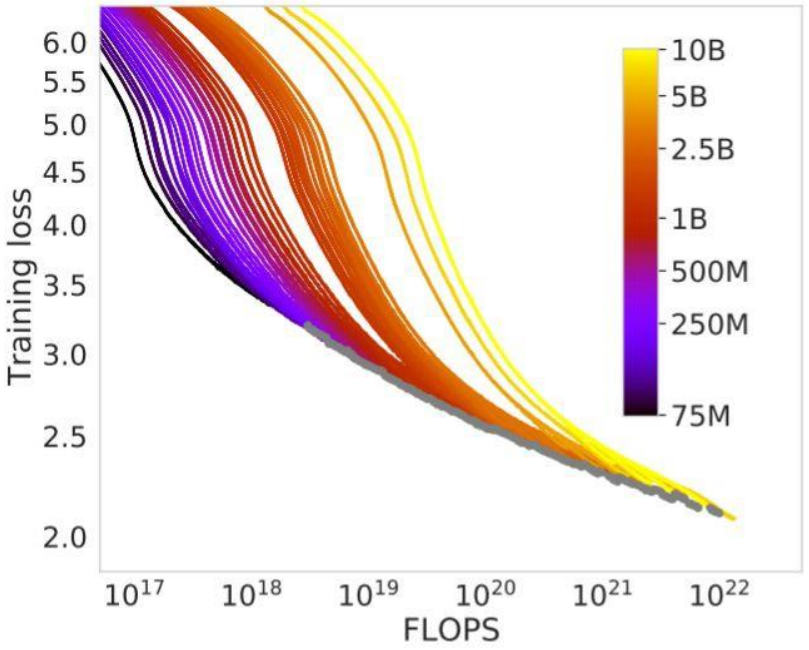
Key Idea

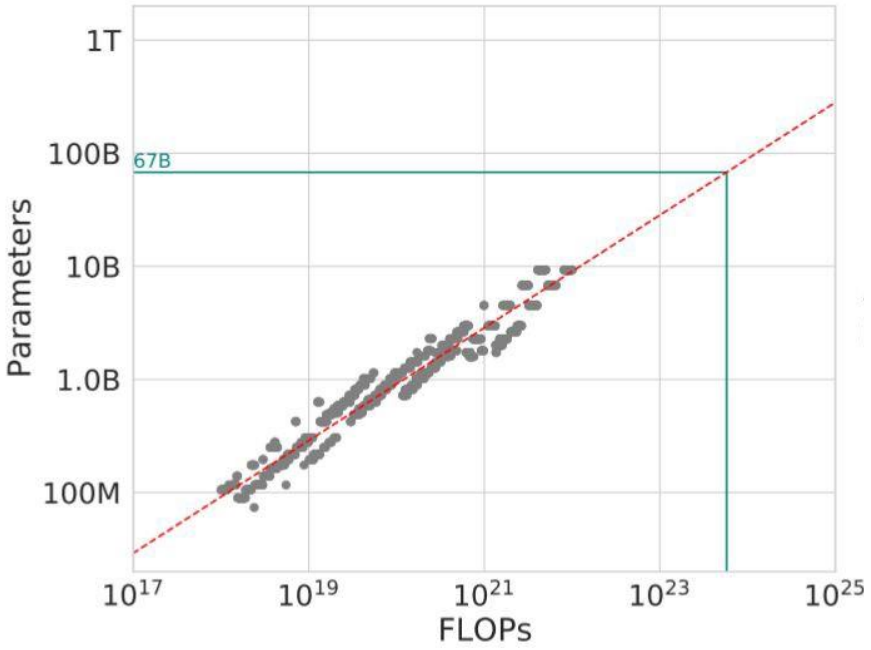

1T
100B
10B
1.0B
100M
10M
Parameters
10^{17}
10^{19}
10^{21}
10^{23}
10^{25}
FLOPs
Approach 1
Approach 2
Approach 3
Kaplan et al (2020)
Chinchilla (70B)
Gopher (280B)
GPT-3 (175B)
Megatron-Turing NLG (530B)

Approach 1: Fix model sizes & vary number of training tokens

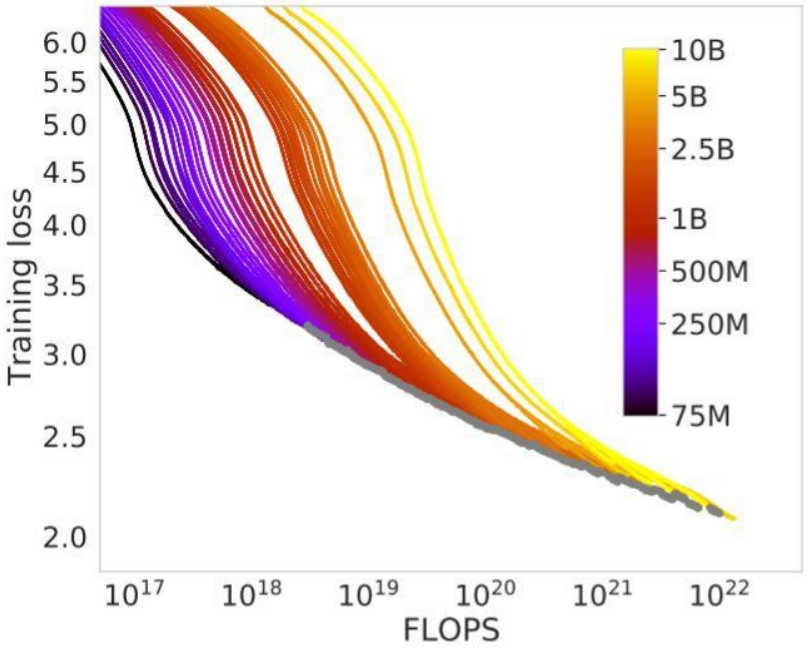


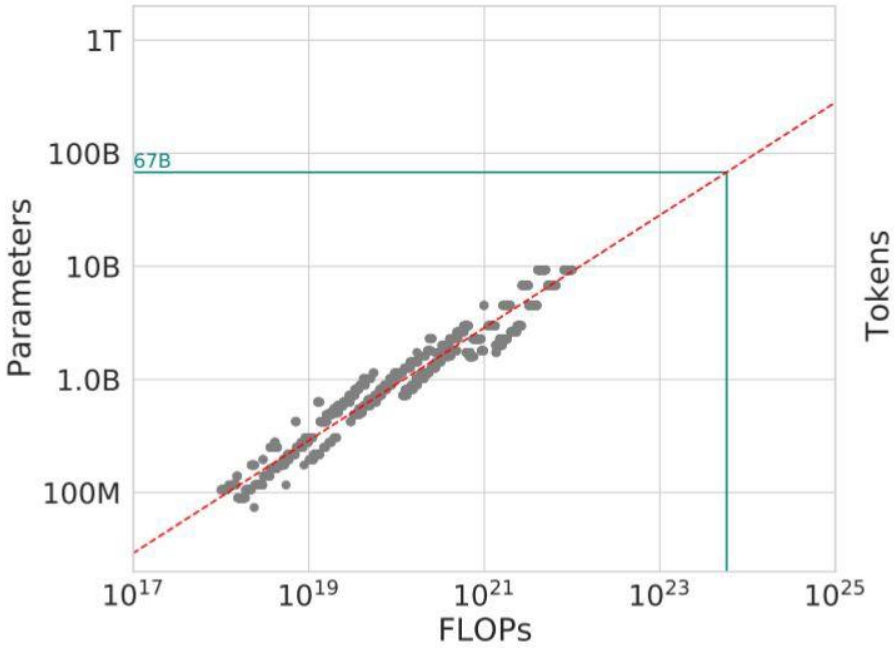
Approach 1: Minimum over training curves

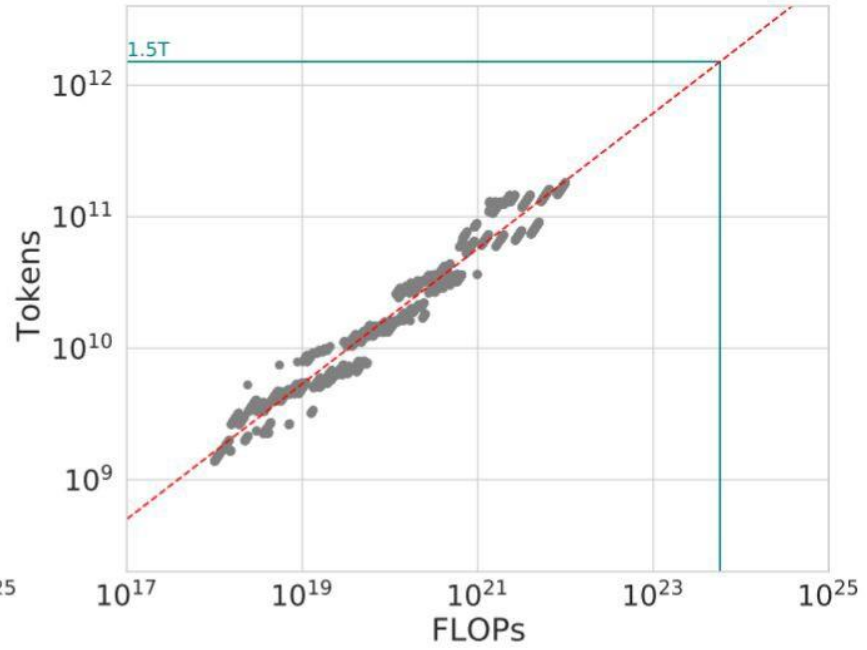




Approach 1: Minimum over training curves

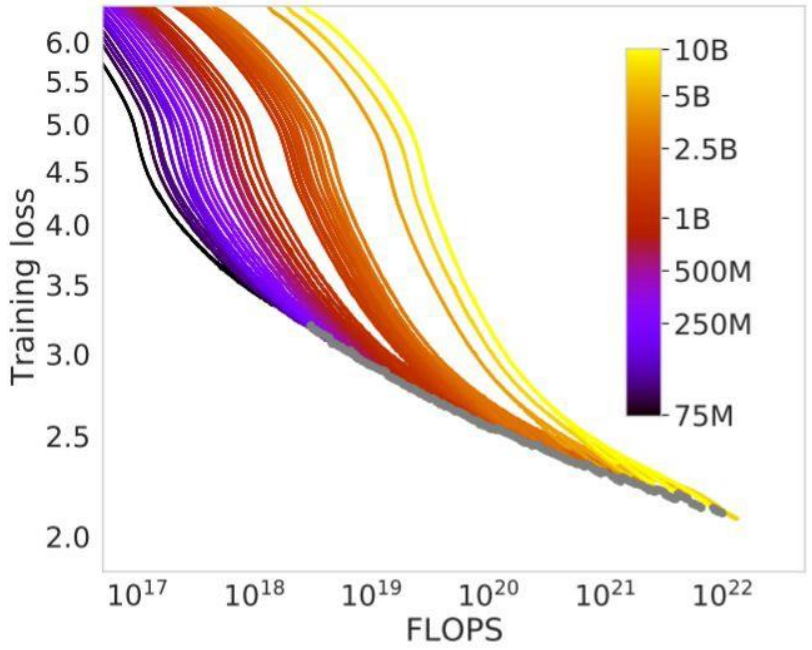


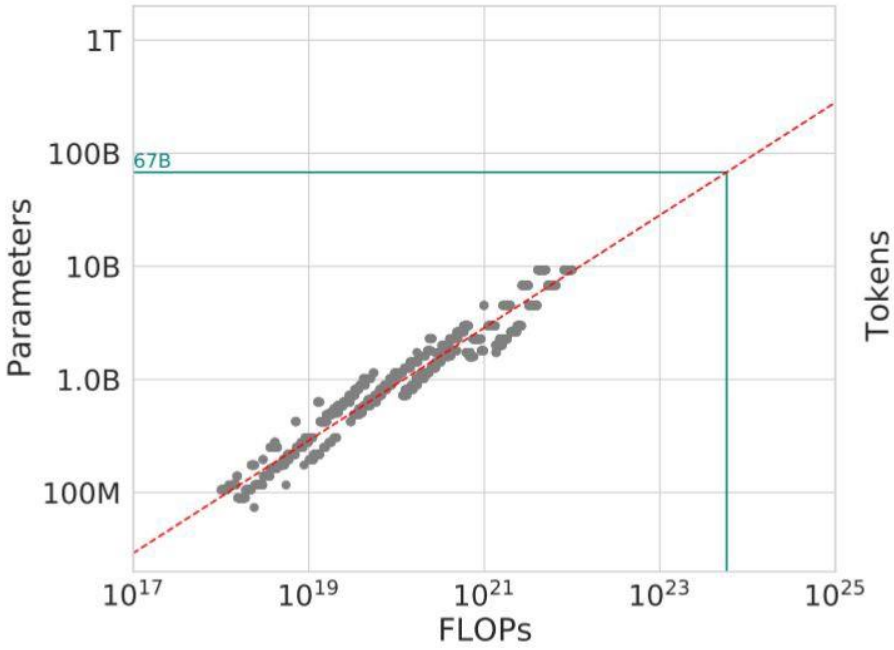


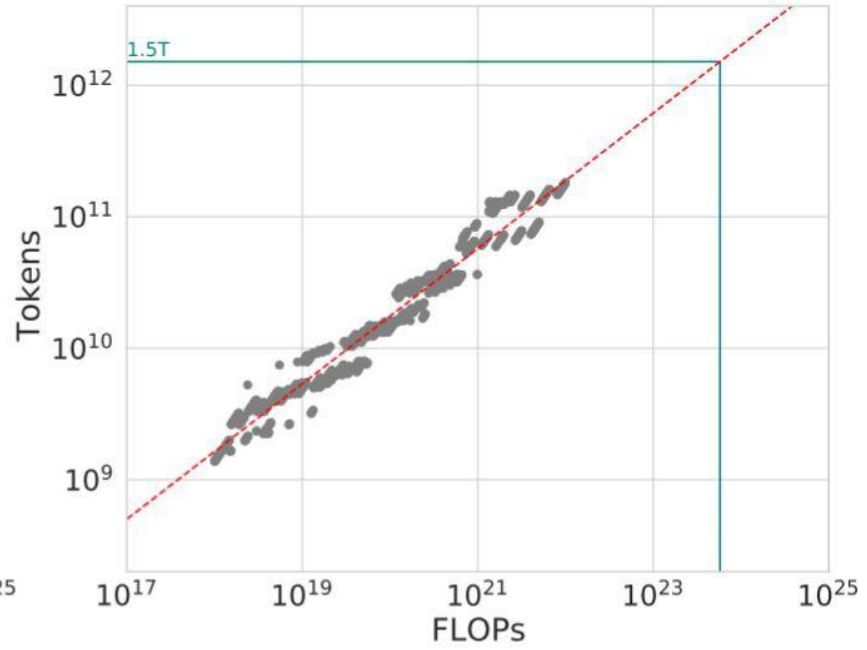


$$C \approx 6ND$$

Approach 1: Minimum over training curves

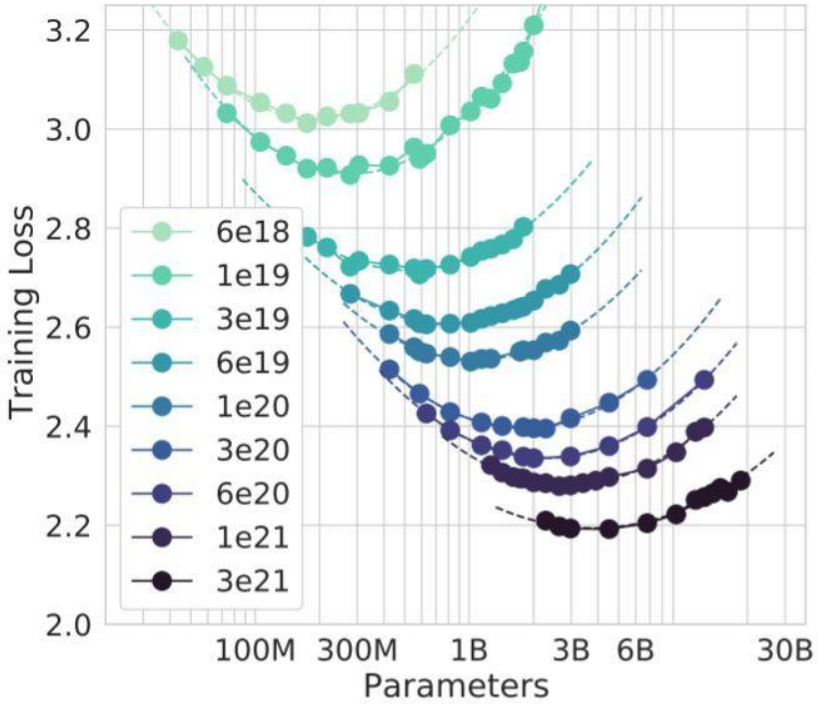






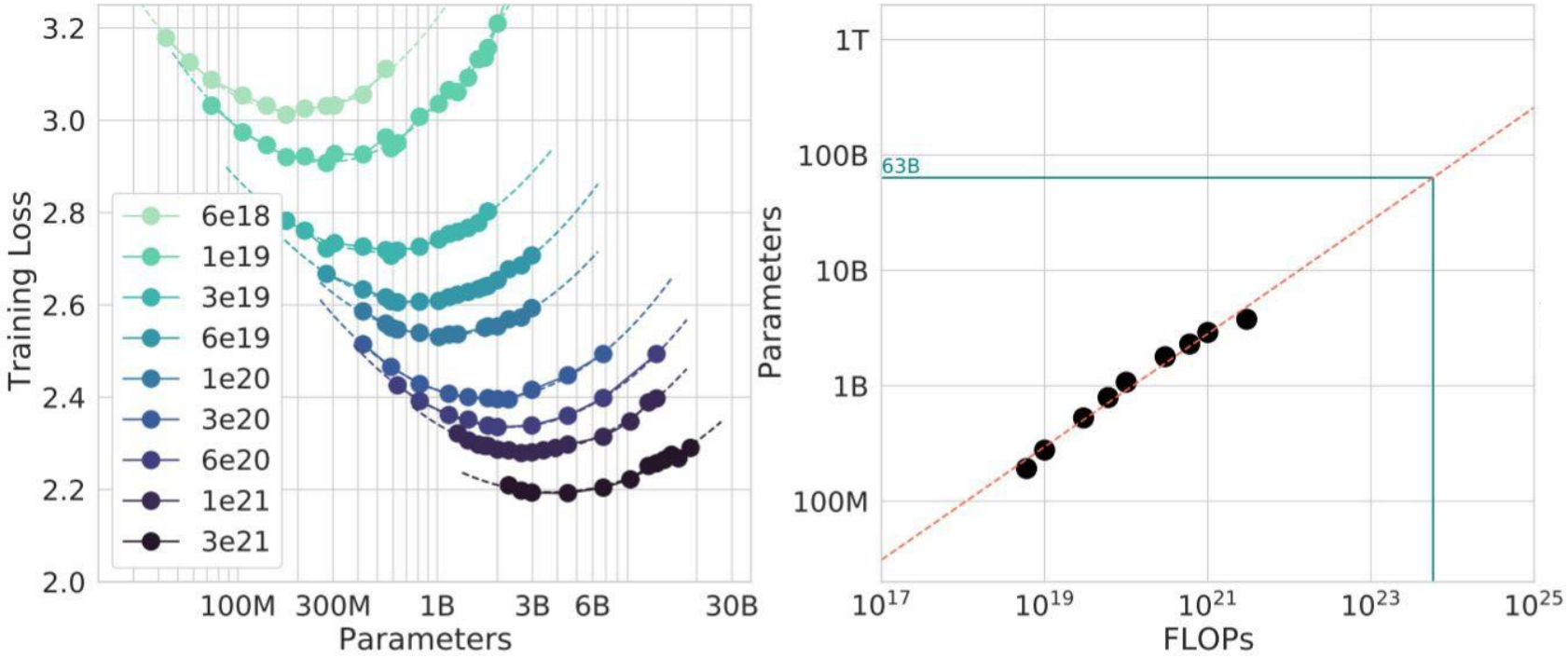
Approach	Coeff. a where $N_{opt} \propto C^a$	Coeff. b where $D_{opt} \propto C^b$
1. Minimum over training curves	0.50 (0.488, 0.502)	0.50 (0.501, 0.512)
Kaplan et al. (2020)	0.73	0.27

Approach 2: IsoFLOP Profiles

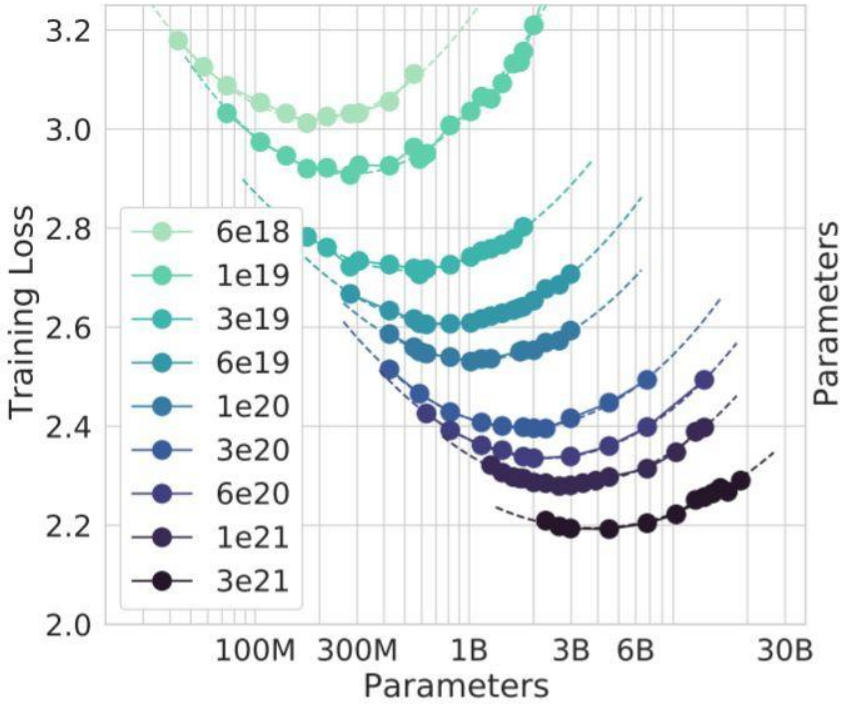


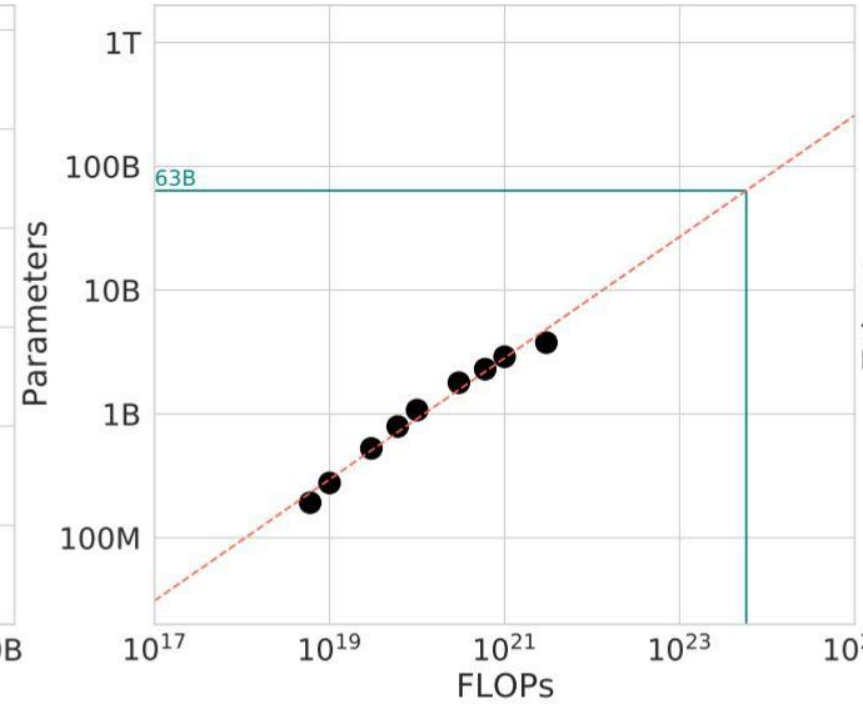
$$C \approx 6ND$$

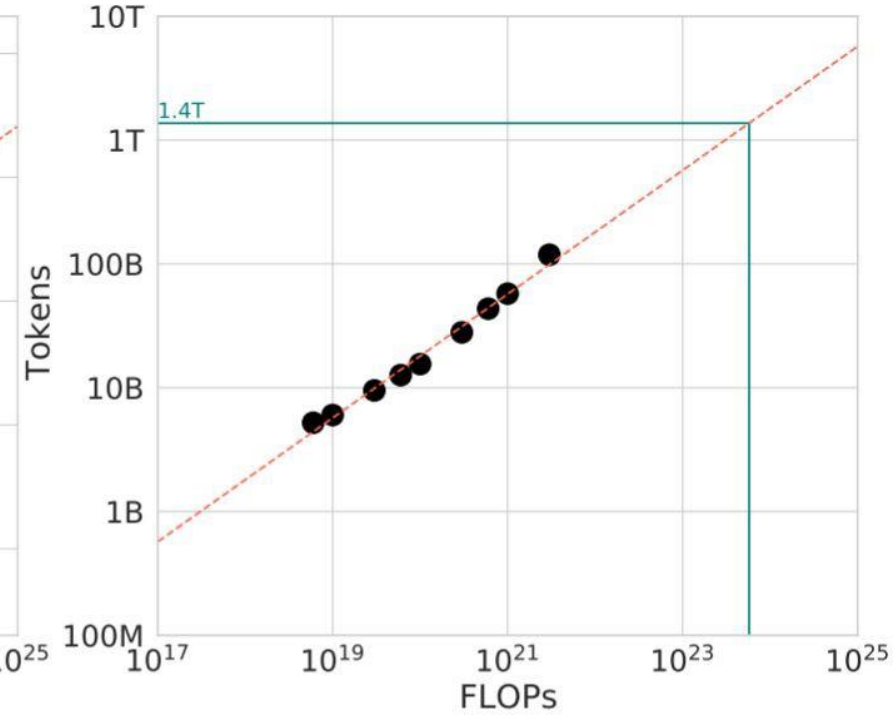
Approach 2: IsoFLOP Profiles



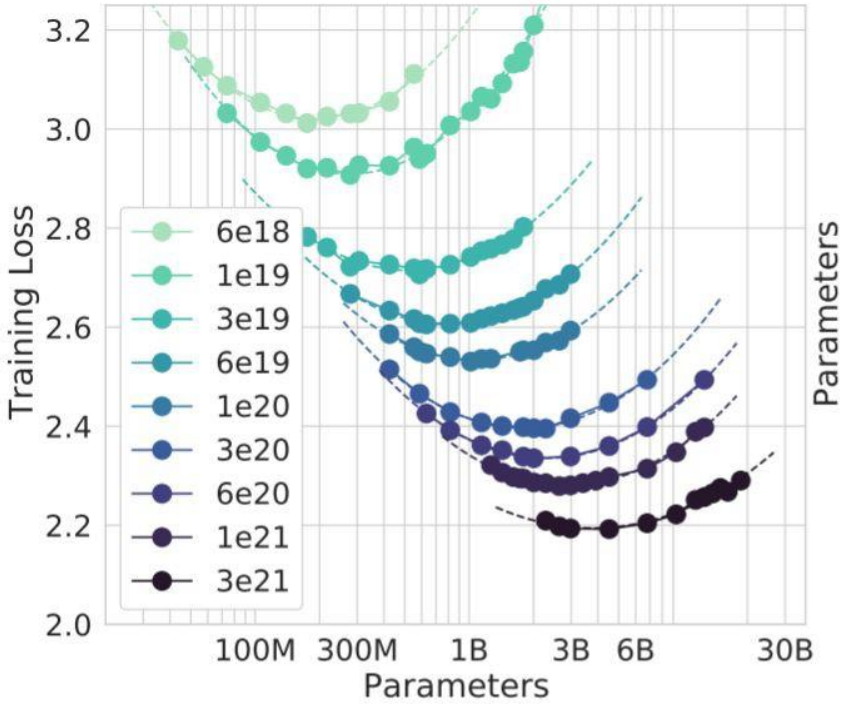
Approach 2: IsoFLOP Profiles

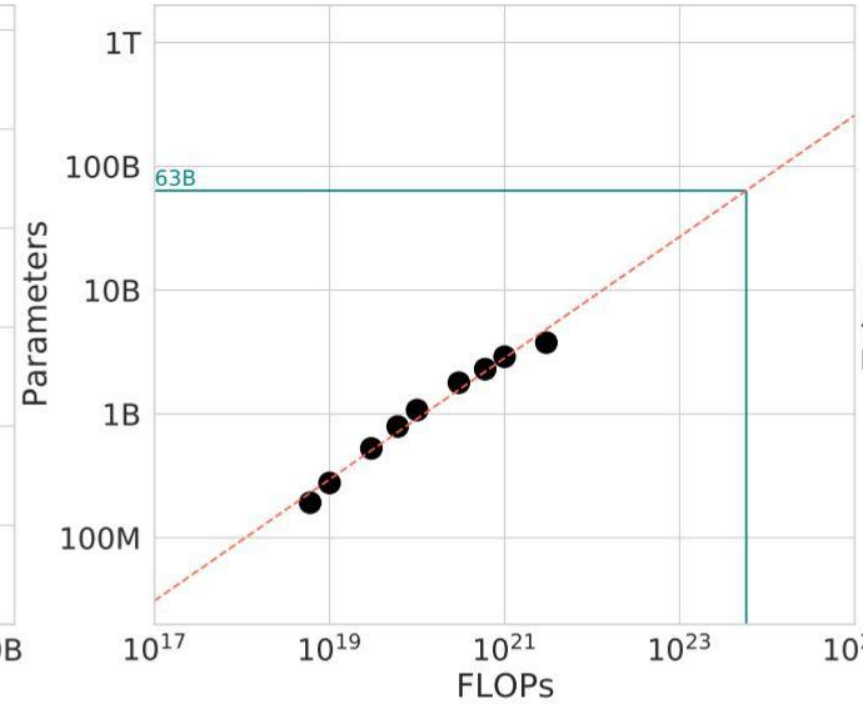


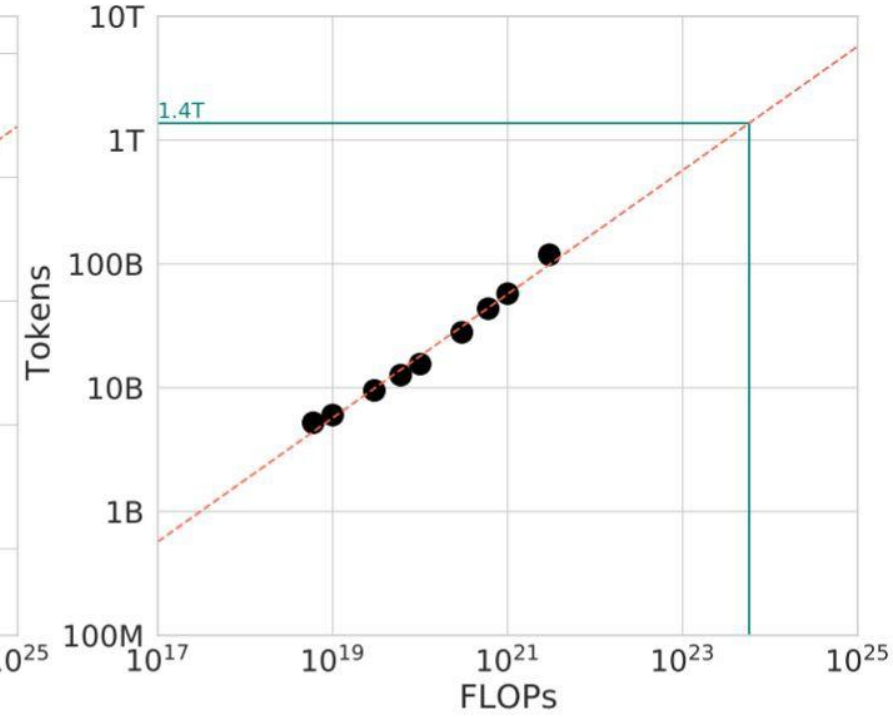




Approach 2: IsoFLOP Profiles







Approach	Coeff. a where $N_{opt} \propto C^a$	Coeff. b where $D_{opt} \propto C^b$
1. Minimum over training curves	0.50 (0.488, 0.502)	0.50 (0.501, 0.512)
2. IsoFLOP profiles	0.49 (0.462, 0.534)	0.51 (0.483, 0.529)
Kaplan et al. (2020)	0.73	0.27

Approach 3: Fitting a Parametric Function

$$\hat{L}(N, D) \triangleq E + \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}}$$

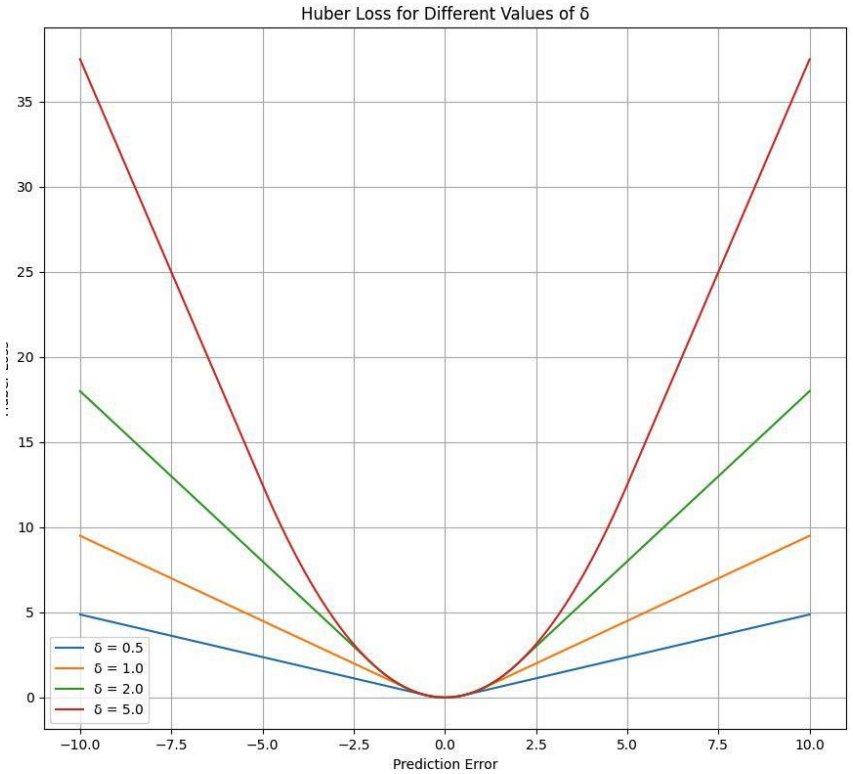
Entropy of natural text (minimum possible loss for any generative process)

Transformer under-performs ideal generative process

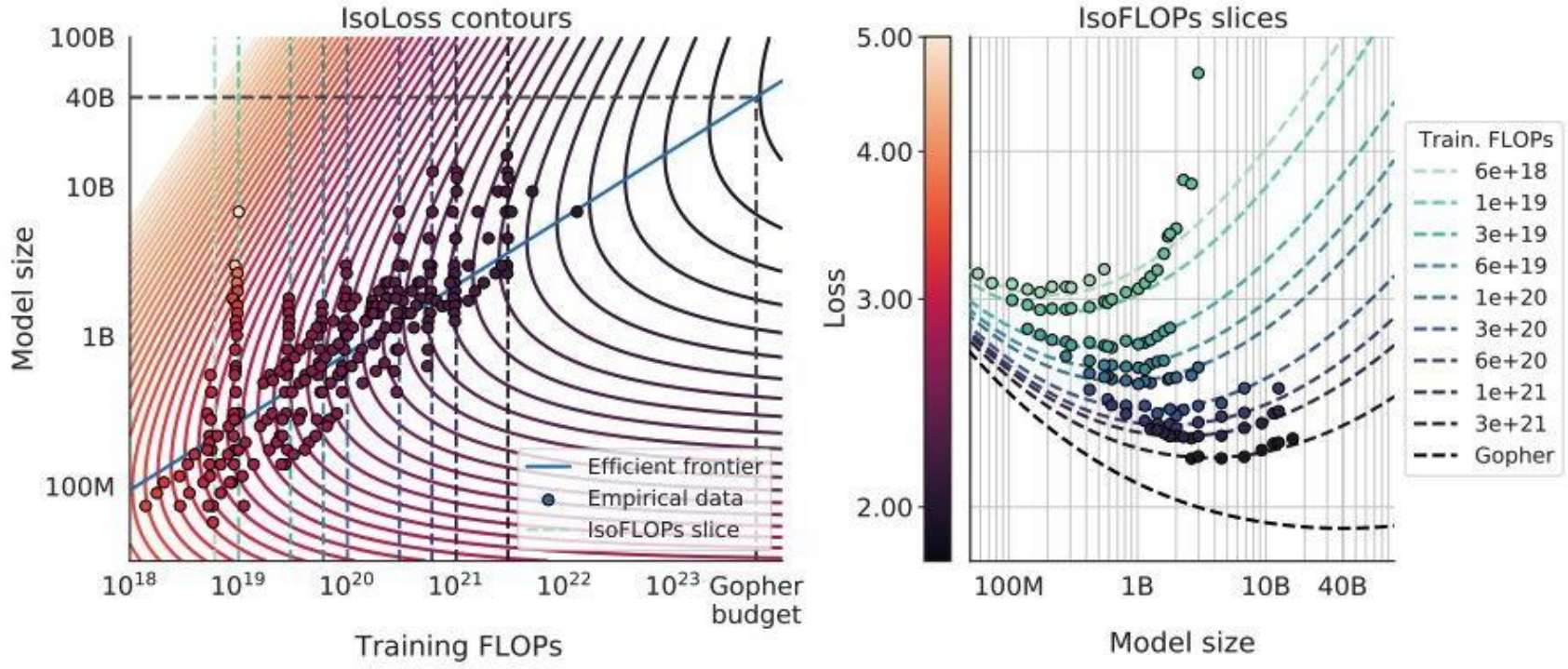
Transformer is not trained to convergence

Approach 3: Fitting a Parametric Function

$$\hat{L}(N, D) \triangleq E + \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}} \qquad \min_{A,B,E,\alpha,\beta} \sum_{\text{Runs } i} \text{Huber}_{\delta}\left(\log \hat{L}(N_i, D_i) - \log L_i\right)$$



Approach 3: Fitting a Parametric Function



Approach	Coeff. a where $N_{opt} \propto C^a$	Coeff. b where $D_{opt} \propto C^b$
1. Minimum over training curves	0.50 (0.488, 0.502)	0.50 (0.501, 0.512)
2. IsoFLOP profiles	0.49 (0.462, 0.534)	0.51 (0.483, 0.529)
3. Parametric modelling of the loss	0.46 (0.454, 0.455)	0.54 (0.542, 0.543)
Kaplan et al. (2020)	0.73	0.27

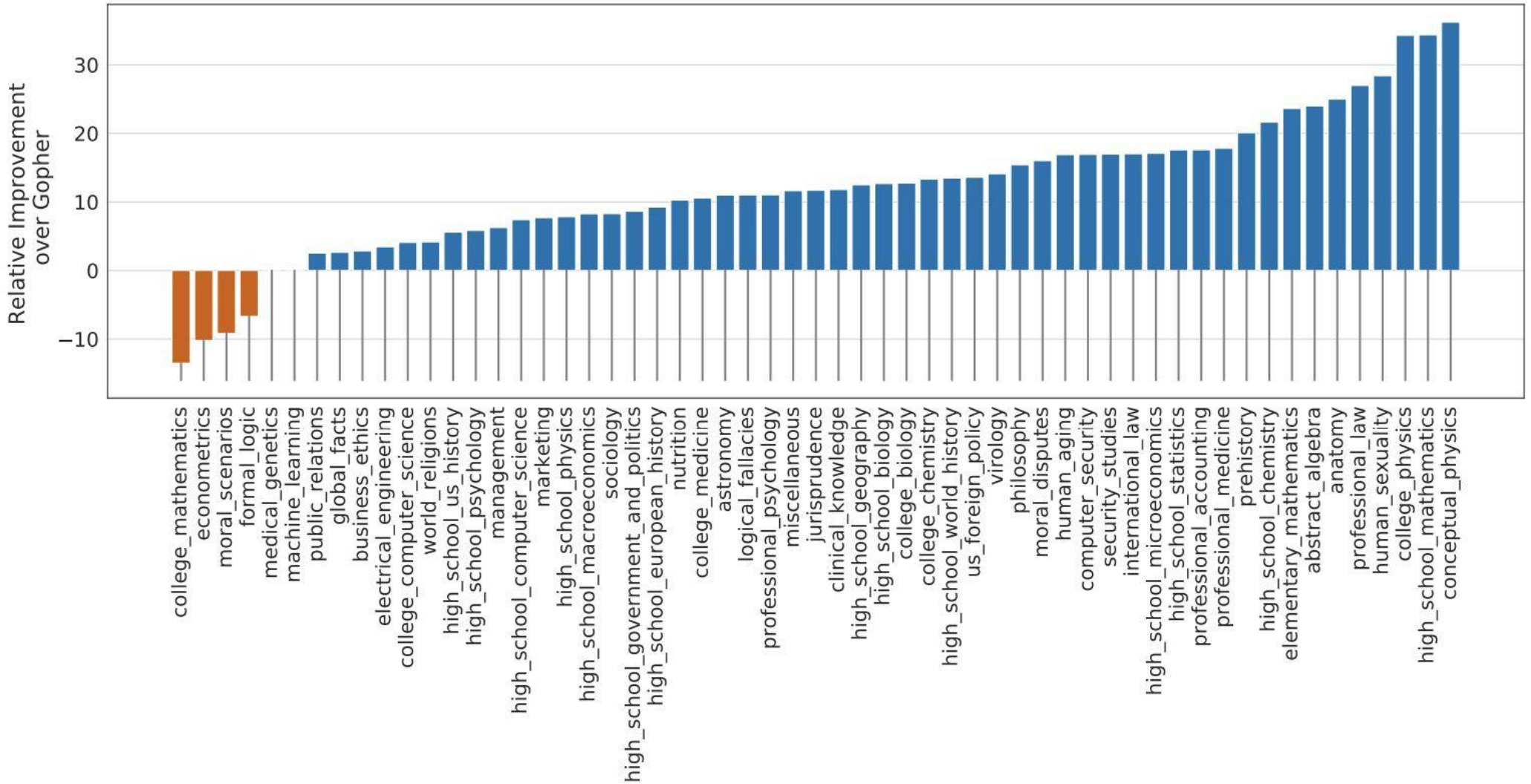
Optimal Model Scaling

- Gopher LLM: 280B parameters, trained on 300B tokens. **Non optimal**
- Test this hypothesis by training a compute-optimal version of the Gopher LLM, using the same number of FLOPs

Parameters	FLOPs	FLOPs (in *Gopher* unit)	Tokens
400 Million	1.92e+19	1/29,968	8.0 Billion
1 Billion	1.21e+20	1/4,761	20.2 Billion
10 Billion	1.23e+22	1/46	205.1 Billion
67 Billion	5.76e+23	1	1.5 Trillion
175 Billion	3.85e+24	6.7	3.7 Trillion
280 Billion	9.90e+24	17.2	5.9 Trillion
520 Billion	3.43e+25	59.5	11.0 Trillion
1 Trillion	1.27e+26	221.3	21.2 Trillion
10 Trillion	1.30e+28	22515.9	216.2 Trillion

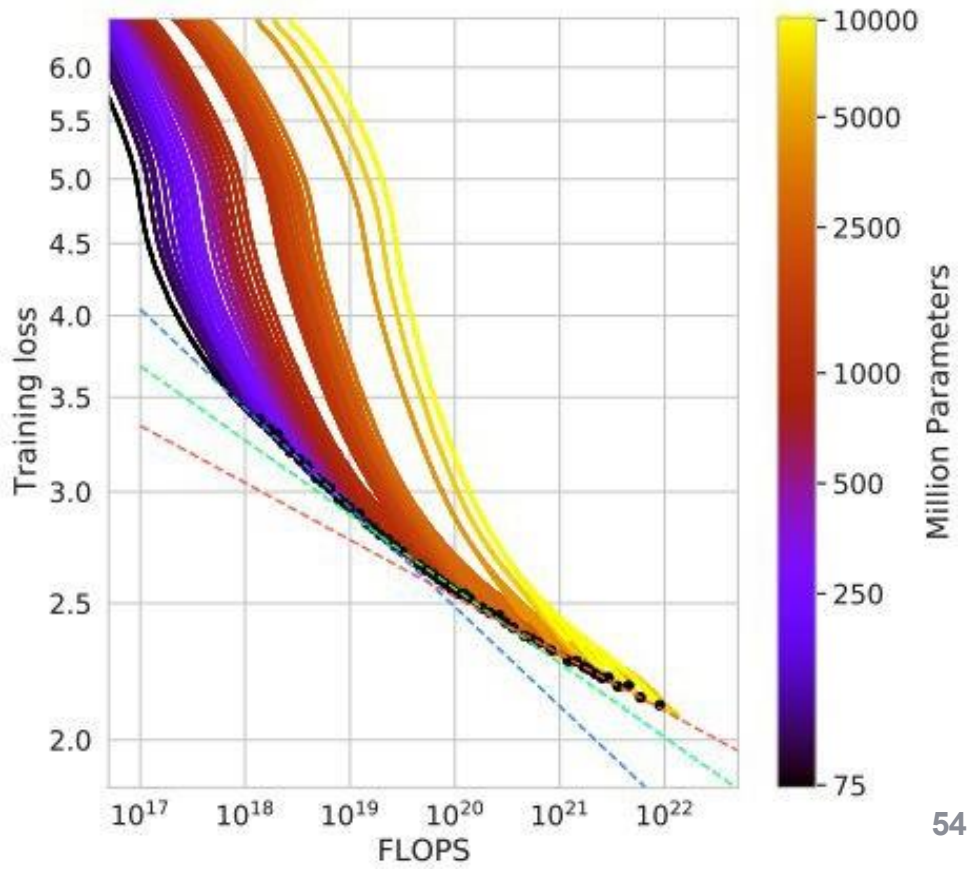
Rae, J. W., et al. "van den, Hendricks, LA, Rauh, M., Huang, P.-S.,... Irving, G.(2022)." *Scaling language*.

Chinchilla 70B vs Gopher 280B



Discussion and Conclusion

- Parameters and tokens should be increased equally
- Models circa chinchilla were oversized



Questions from you

- Do you find the human evaluation results in this paper convincing and why?
- So my question is how effectively can we adapt this RM + PPO alignment pipeline for video generation or multi-model stuff ..

ImageReward: Learning and Evaluating Human Preferences for Text-to-Image Generation

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, Yuxiao Dong

We present a comprehensive solution to learn and improve text-to-image models from human preference feedback. To begin with, we build ImageReward -- the first general-purpose text-to-image human preference reward model -- to effectively encode human preferences. Its training is based on our systematic annotation pipeline including rating and ranking, which collects 137k expert comparisons to date. In human evaluation, ImageReward outperforms existing scoring models and metrics, making it a promising automatic metric for evaluating text-to-image synthesis. On top of it, we propose Reward Feedback Learning (ReFL), a direct tuning algorithm to optimize diffusion models against a scorer. Both automatic and human evaluation support ReFL's advantages over compared methods. All code and datasets are provided at \url{this https URL}.

Improving Video Generation with Human Feedback

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, Xiaohong Liu, Fei Yang, Pengfei Wan, Di Zhang, Kun Gai, Yujiu Yang, Wanli Ouyang

Video generation has achieved significant advances through rectified flow techniques, but issues like unsmooth motion and misalignment between videos and prompts persist. In this work, we develop a systematic pipeline that harnesses human feedback to mitigate these problems and refine the video generation model. Specifically, we begin by constructing a large-scale human preference dataset focused on modern video generation models, incorporating pairwise annotations across multi-dimensions. We then introduce VideoReward, a multi-dimensional video reward model, and examine how annotations and various design choices impact its rewarding efficacy. From a unified reinforcement learning perspective aimed at maximizing reward with KL regularization, we introduce three alignment algorithms for flow-based models. These include two training-time strategies: direct preference optimization for flow (Flow-DPO) and reward weighted regression for flow (Flow-RWR), and an inference-time technique, Flow-NRG, which applies reward guidance directly to noisy videos. Experimental results indicate that VideoReward significantly outperforms existing reward models, and Flow-DPO demonstrates superior performance compared to both Flow-RWR and supervised fine-tuning methods. Additionally, Flow-NRG lets users assign custom weights to multiple objectives during inference, meeting personalized video quality needs.

Visual autoregressive modeling: Scalable image generation via next-scale prediction

K Tian, Y Jiang, Z Yuan, B Peng, L Wang

Advances in neural information processing systems, 2024 • proceedings.neurips.cc

Abstract

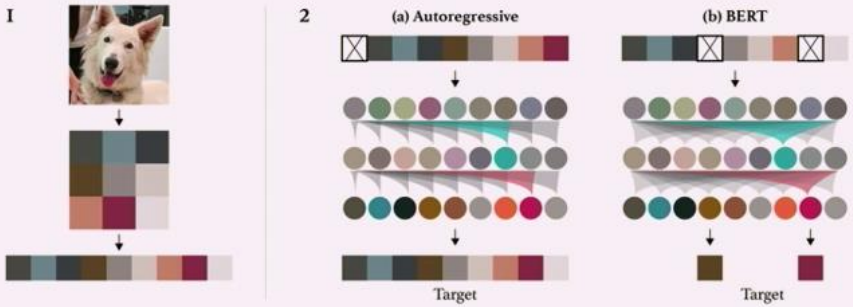
We present Visual AutoRegressive modeling (VAR), a new generation paradigm that redefines the autoregressive learning on images as coarse-to-fine" next-scale prediction" or" next-resolution prediction", diverging from the standard raster-scan" next-token prediction". This simple, intuitive methodology allows autoregressive (AR) transformers to learn visual distributions fast and generalize well: VAR, for the first time, makes GPT-style AR models surpass diffusion transformers in image generation. On ImageNet 256x256

SHOW MORE

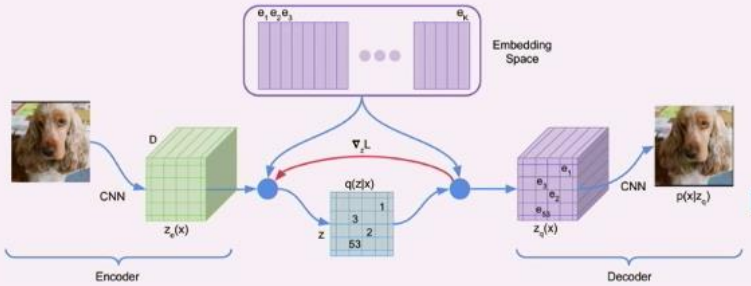
☆ Save Cite Cited by 1195 Related articles All 8 versions »

*Acknowledgement: Many of the following slides are modified from Yi Jiang

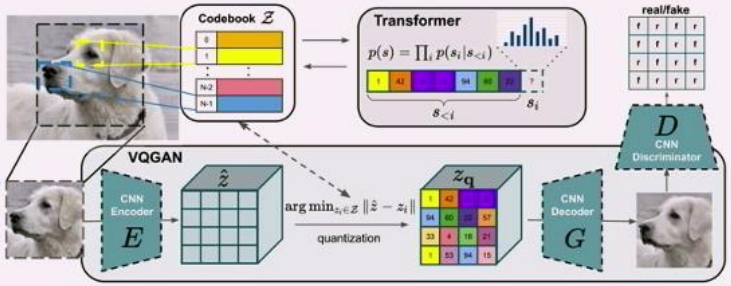
Image Tokenizer and AutoRegressive Transformers



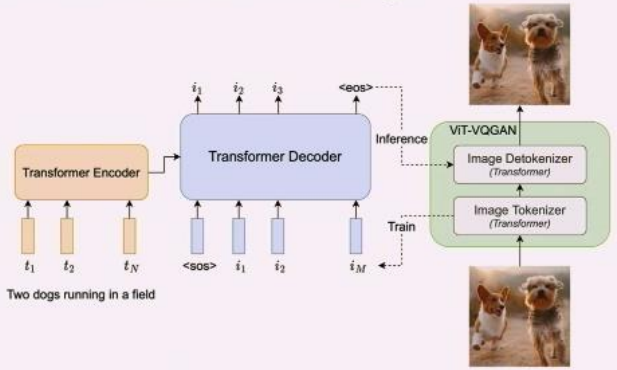
iGPT: Generative Image & Pretraining from Pixels



VQVAE: Tokenize Images into discrete token index



VQGAN: Image tokenizer + AutoRegressive transformer



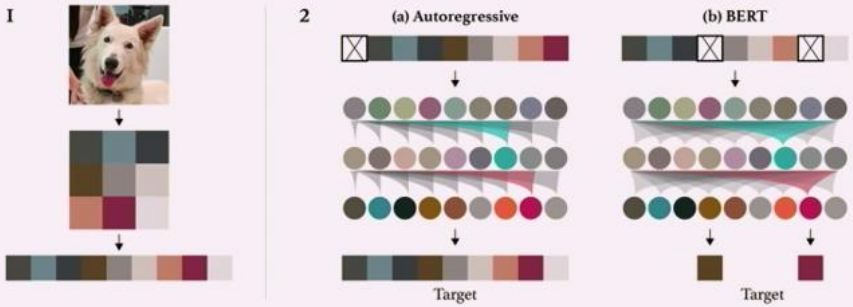
Parti : Scaling Up the AutoRegressive transformer

[1] Mark Chen et al., Generative Pretraining From Pixels, in ICML, 2020.
[2] Aaron van et al., Neural Discrete Representation Learning, 2017.
[3] PATRICK ESSER et al., Taming Transformers for High-Resolution Image Synthesis, 2021 CVPR.
[4] Jiahui Yu et al., Scaling Autoregressive Models for Content-Rich Text-to-Image Generation, 2022 Arxiv.

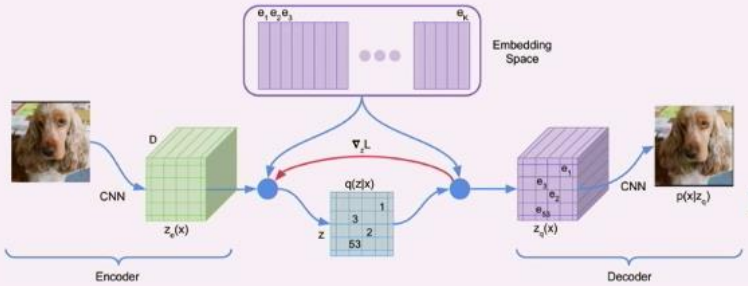
Image Tokenizer and AutoRegressive Transformers

Challenges:

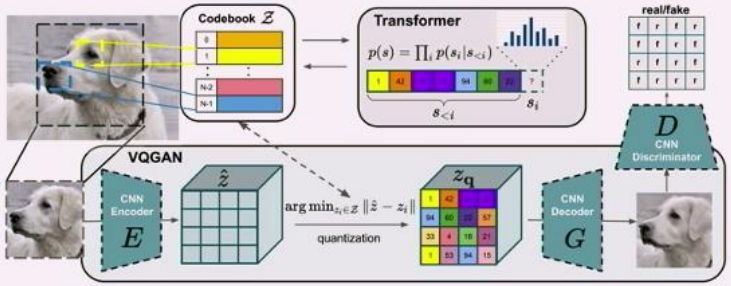
- **Autoregressive models perform significantly worse than sota diffusion models in high-resolution image systhesis.**



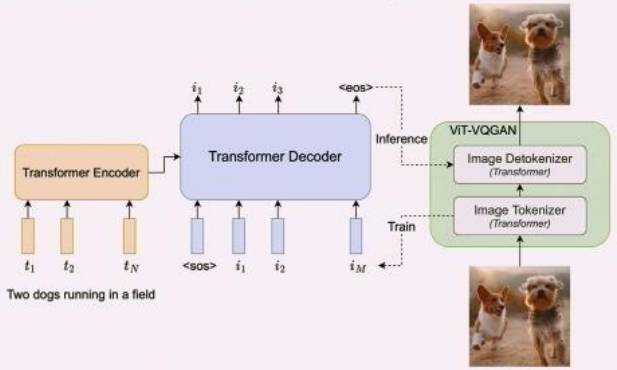
iGPT: Generative Image & Pretraining from Pixels



VQVAE: Tokenize Images into discrete token index



VQGAN: Image tokenizer + AutoRegressive transformer



Parti : Scaling Up the AutoRegressive transformer

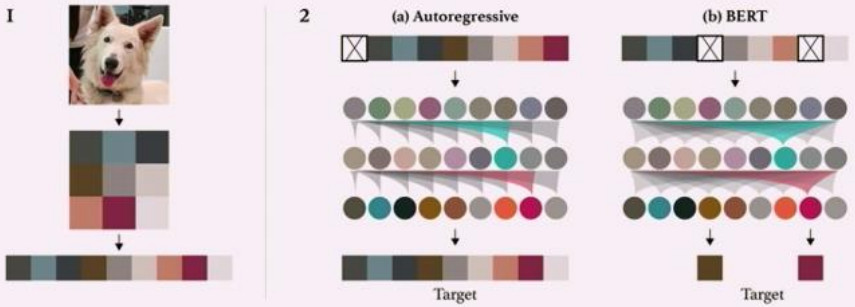
[1] Mark Chen et al., Generative Pretraining From Pixels, in ICML, 2020.

[2] Aaron van et al., Neural Discrete Representation Learning, 2017.

[3] PATRICK ESSER et al., Taming Transformers for High-Resolution Image Synthesis, 2021 CVPR.

[4] Jiahui Yu et al., Scaling Autoregressive Models for Content-Rich Text-to-Image Generation, 2022 Arxiv.

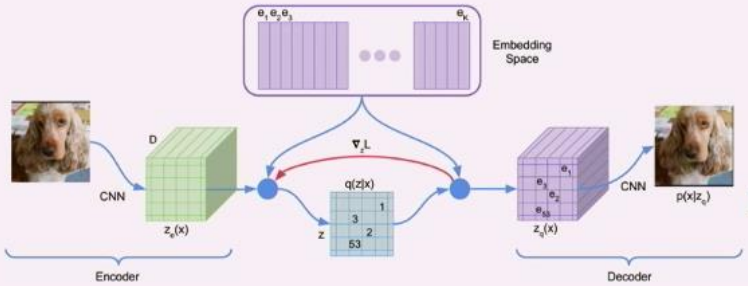
Image Tokenizer and AutoRegressive Transformers



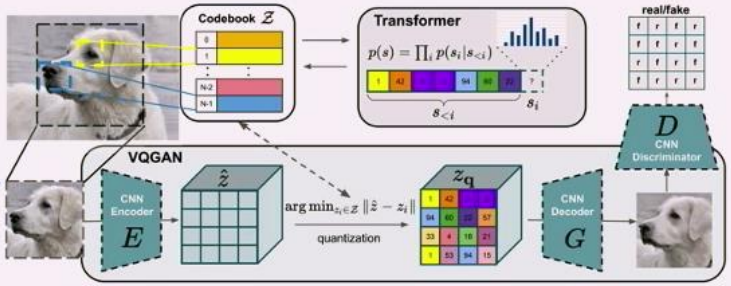
iGPT: Generative Image & Pretraining from Pixels

Challenges:

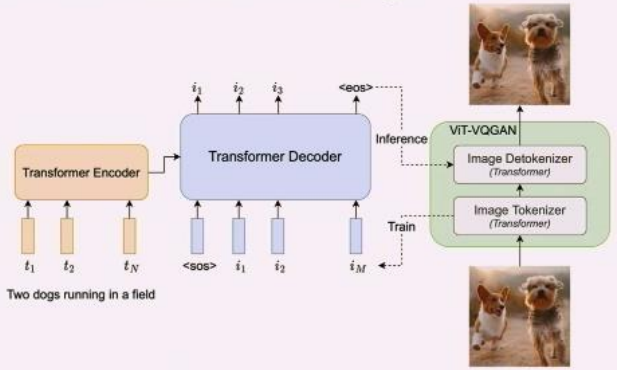
- **Autoregressive models perform significantly worse than sota diffusion models in high-resolution image systhesis.**
- **Autoregressive models have not demonstrated the same scaling law properties as LLMs in text generation.**



VQVAE: Tokenize Images into discrete token index



VQGAN: Image tokenizer + AutoRegressive transformer



Parti : Scaling Up the AutoRegressive transformer

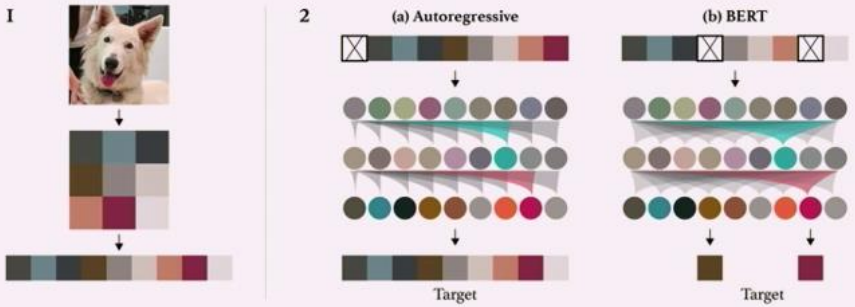
[1] Mark Chen et al., Generative Pretraining From Pixels, in ICML, 2020.

[2] Aaron van et al., Neural Discrete Representation Learning, 2017.

[3] PATRICK ESSER et al., Taming Transformers for High-Resolution Image Synthesis, 2021 CVPR.

[4] Jiahui Yu et al., Scaling Autoregressive Models for Content-Rich Text-to-Image Generation, 2022 Arxiv.

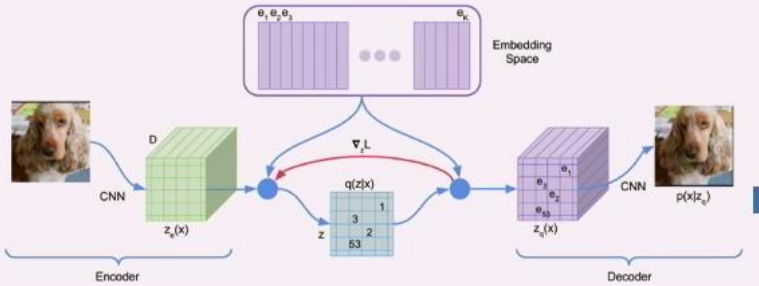
Image Tokenizer and AutoRegressive Transformers



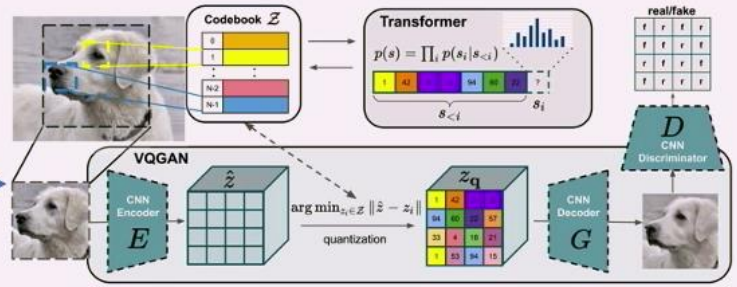
iGPT: Generative Image & Pretraining from Pixels

Challenges:

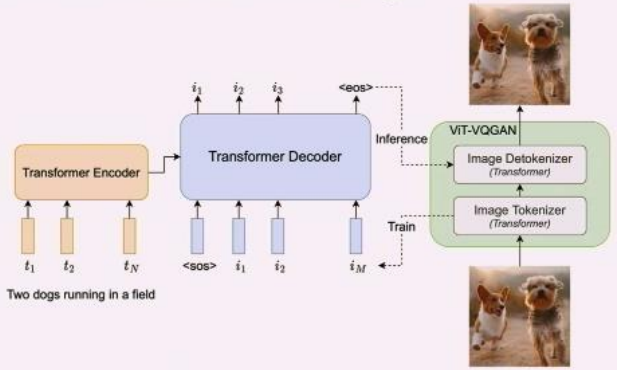
- **Autoregressive models perform significantly worse than sota diffusion models in high-resolution image systhesis.**
- **Autoregressive models have not demonstrated the same scaling law properties as LLMs in text generation.**
- **Due to raster order prediction, autoregressive models suffer from very slow prediction speeds.**



VQVAE: Tokenize Images into discrete token index



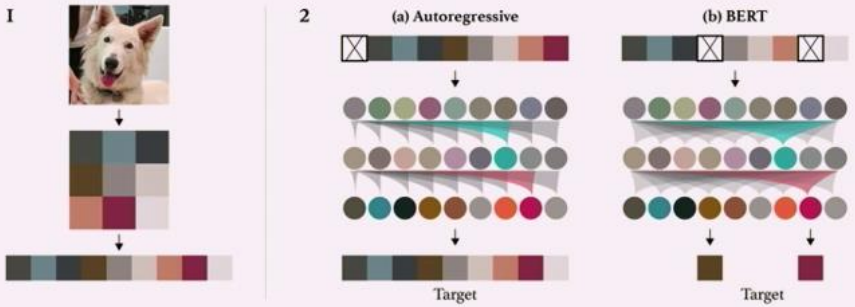
VQGAN: Image tokenizer + AutoRegressive transformer



Parti : Scaling Up the AutoRegressive transformer

[1] Mark Chen et al., Generative Pretraining From Pixels, in ICML, 2020.
[2] Aaron van et al., Neural Discrete Representation Learning, 2017.
[3] PATRICK ESSER et al., Taming Transformers for High-Resolution Image Synthesis, 2021 CVPR.
[4] Jiahui Yu et al., Scaling Autoregressive Models for Content-Rich Text-to-Image Generation, 2022 Arxiv.

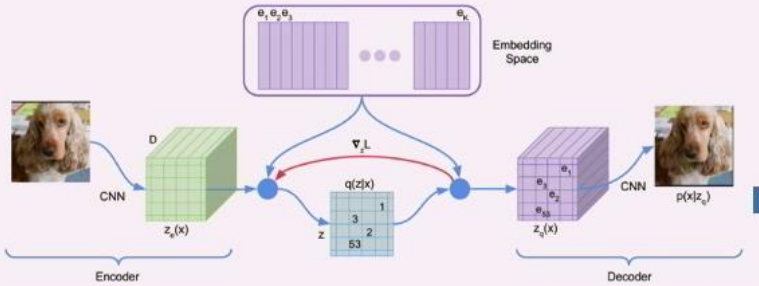
Image Tokenizer and AutoRegressive Transformers



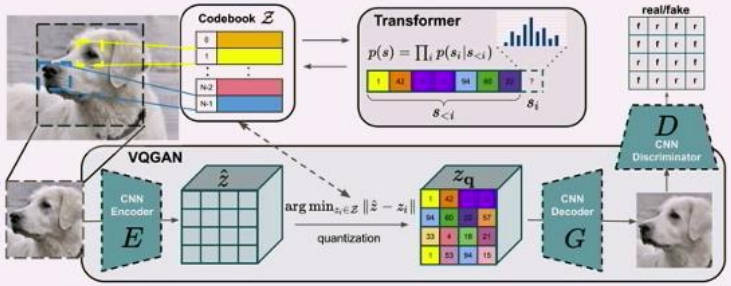
iGPT: Generative Image & Pretraining from Pixels

Challenges:

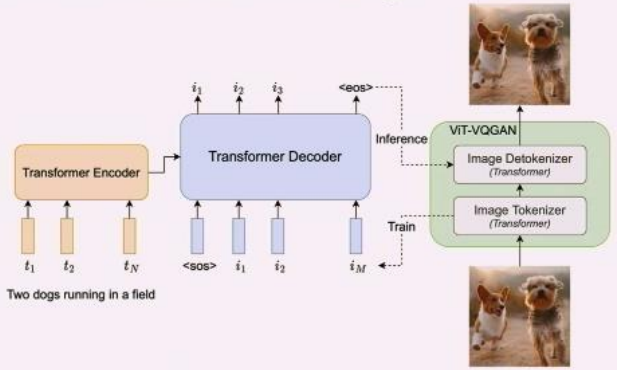
- **Autoregressive models perform significantly worse than sota diffusion models in high-resolution image systhesis.**
- **Autoregressive models have not demonstrated the same scaling law properties as LLMs in text generation.**
- **Due to raster order prediction, autoregressive models suffer from very slow prediction speeds.**
- **The raster-scan order is not the most natural "order" for images, as it loses the global information crucial for visual modeling.**



VQVAE: Tokenize Images into discrete token index



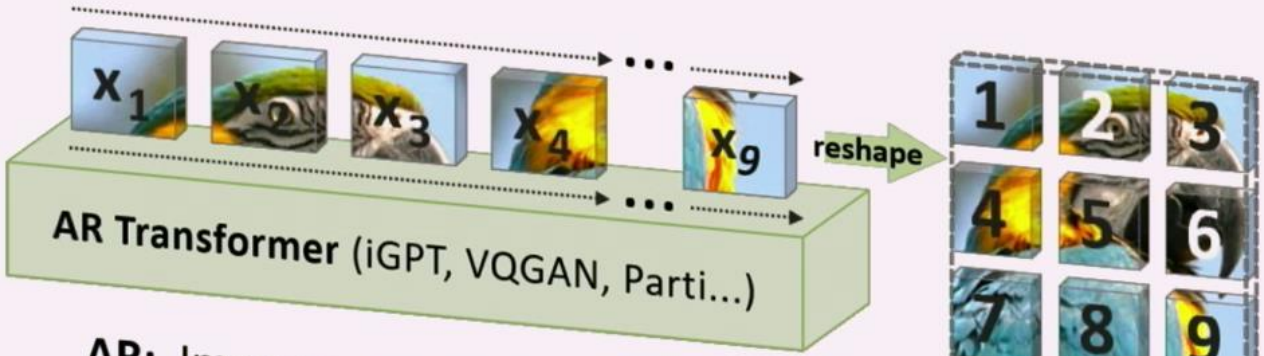
VQGAN: Image tokenizer + AutoRegressive transformer



Parti : Scaling Up the AutoRegressive transformer

[1] Mark Chen et al., Generative Pretraining From Pixels, in ICML, 2020.
[2] Aaron van et al., Neural Discrete Representation Learning, 2017.
[3] PATRICK ESSER et al., Taming Transformers for High-Resolution Image Synthesis, 2021 CVPR.
[4] Jiahui Yu et al., Scaling Autoregressive Models for Content-Rich Text-to-Image Generation, 2022 Arxiv.

Image Tokenizer and AutoRegressive Transformers



AR: Image generation by **next-image-token** prediction

Weakness of vanilla autoregressive models :

- **Mathematical premise violation**: The vqvae token with inter-dependent feature vectors contradicts the unidirectional dependency assumption of autoregressive models.
- **Inability to perform some zero-shot generalization**: The unidirectional nature of image autoregressive modeling restricts their generalizability in tasks requiring bidirectional reasoning.
- **Structural degradation**: The flattening disrupts the spatial locality inherent in image feature maps.
- **Inefficiency:** Generating an image token sequence $x = (x_1, x_2, \ldots, x_{n \times n})$ with a conventional self-attention transformer incurs $O(n^2)$ autoregressive steps and $O(n^6)$ computational cost.

Image Tokenizer and AutoRegressive Transformers

These challenges lead us to ask the following questions:

- Can autoregressive models generate images of comparable or even higher quality than diffusion models?
- Can autoregressive models in visual generation exhibit the same scaling law properties as LLMs, where the generation quality improves with scaling up the model size and compute?
- Can autoregressive models, while **retaining the scaling law properties of LLMs**, adopt a more suitable order for image generation instead of raster-scan order?

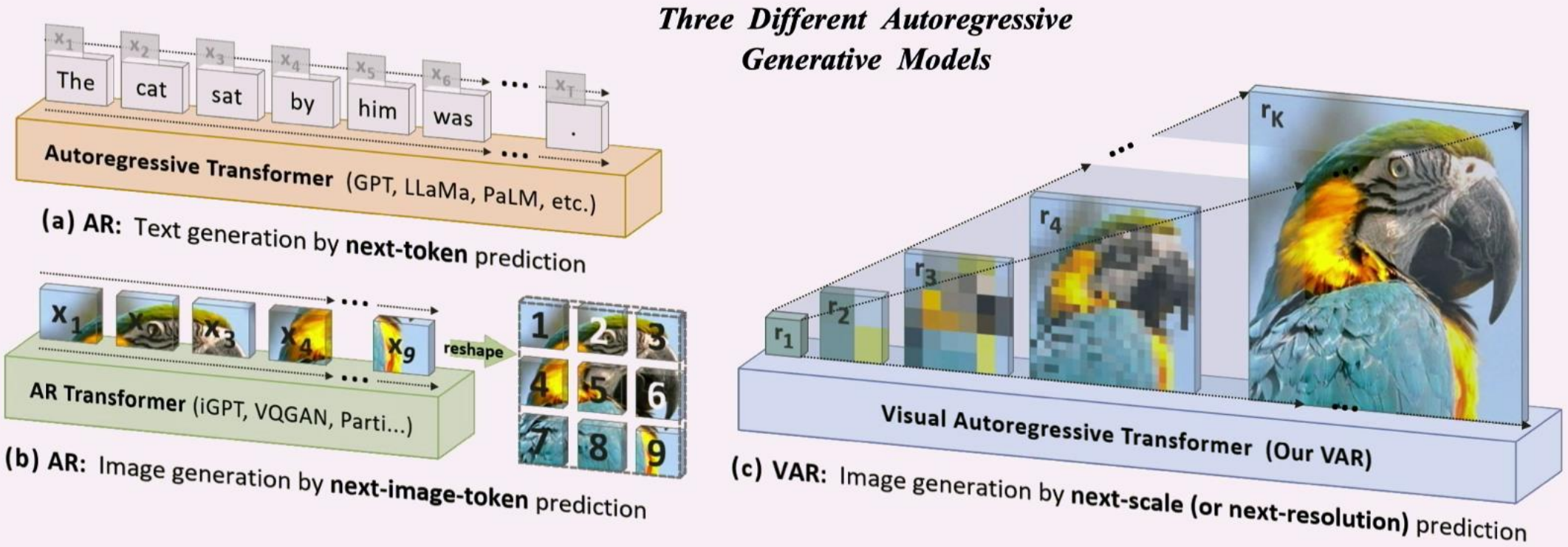
The answer is definitely Yes.

Overview

NEURAL INFORMATION PROCESSING SYSTEMS

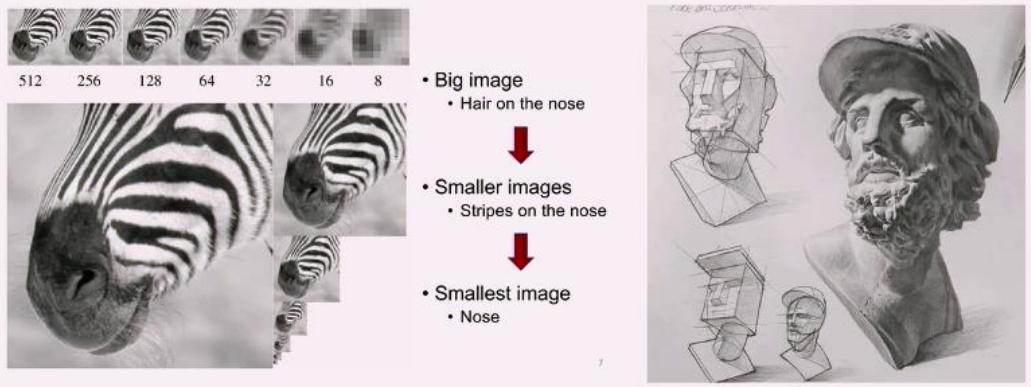
- Generative Models: Diffusion Models & AutoRegressive Models
- Insights from LLM: Tokenization, Next-Token Prediction, Scaling Law
- Image Tokenizer and AutoRegressive Transformers
- **Scalable Visual AutoRegressive Modeling: Next-Scale Prediction**
- Conclusion, discussion, future work

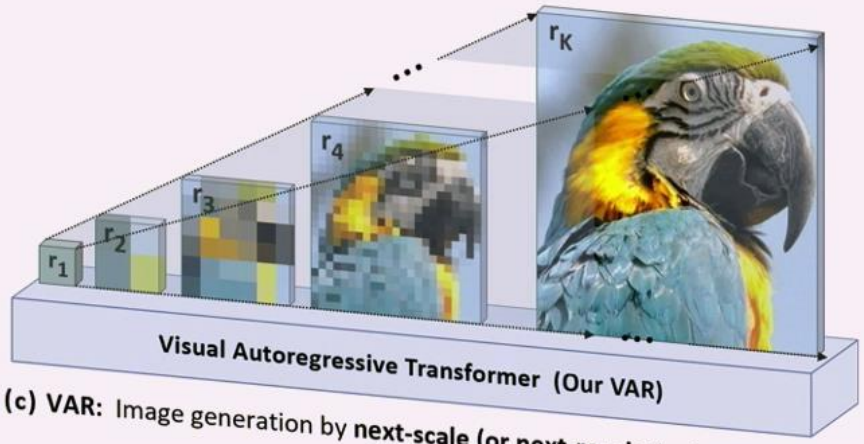
Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction



- **LLMs use an autoregressive approach to predict the next token, as language has a sequential order. However, this method is not suitable for visual data due to its unique characteristics.**
- **Traditional image autoregressive (AR) models use a sequence that is not intuitive to humans but is suitable for computer processing. This sequence predicts image tokens one by one in a top-to-bottom, row-by-row raster order.**
- **Visual AutoRegressive modeling (VAR), a new generation paradigm that redefines the autoregressive learning on images as coarse-to-fine "next-scale prediction"**

Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction

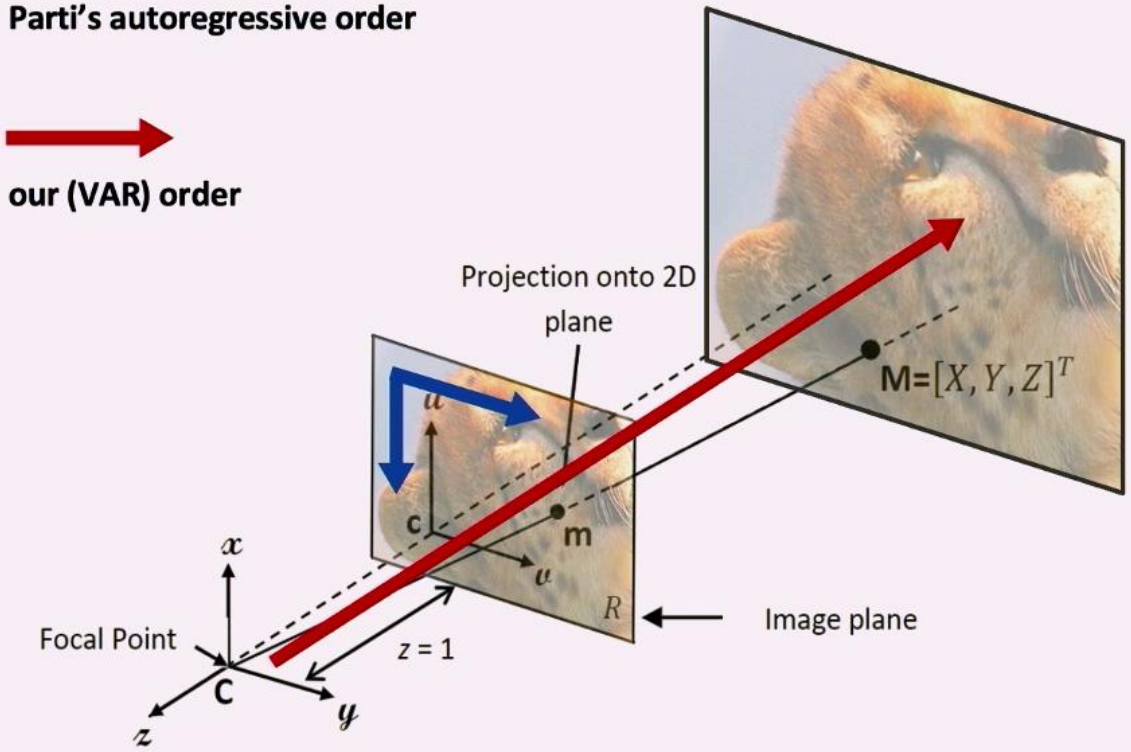




(c) **VAR:** Image generation by **next-scale (or next-resolution)** prediction

Parti's autoregressive order

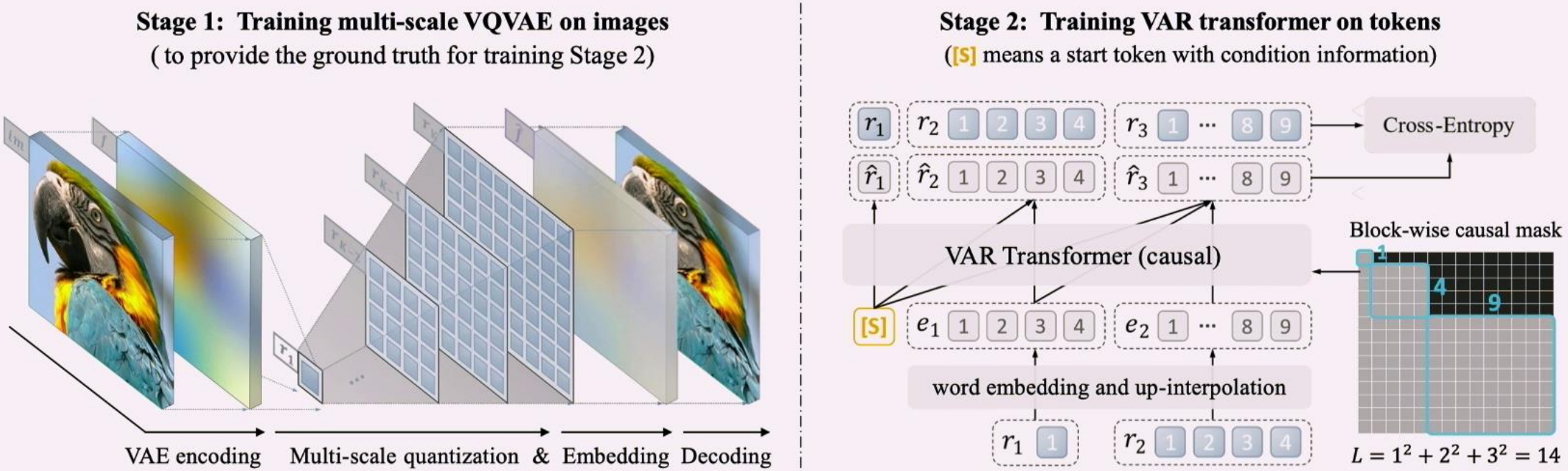
our (VAR) order



Next scale prediction vs Next token prediction

- When humans perceive images or engage in painting, they often start with a holistic overview before delving into finer details.
- This approach of going from coarse to fine, grasping the overall context before refining local details, is very natural.

Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction



- Stage1 : Train multi-scale Image Tokenization for Images, which Tokenize Images into discrete token index
- Stage2: Train **GPT-style(AutoRegressive in scale space)** transformers on VQ-Tokens with teacher forcing

Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction

Algorithm 1: Multi-scale VQVAE Encoding

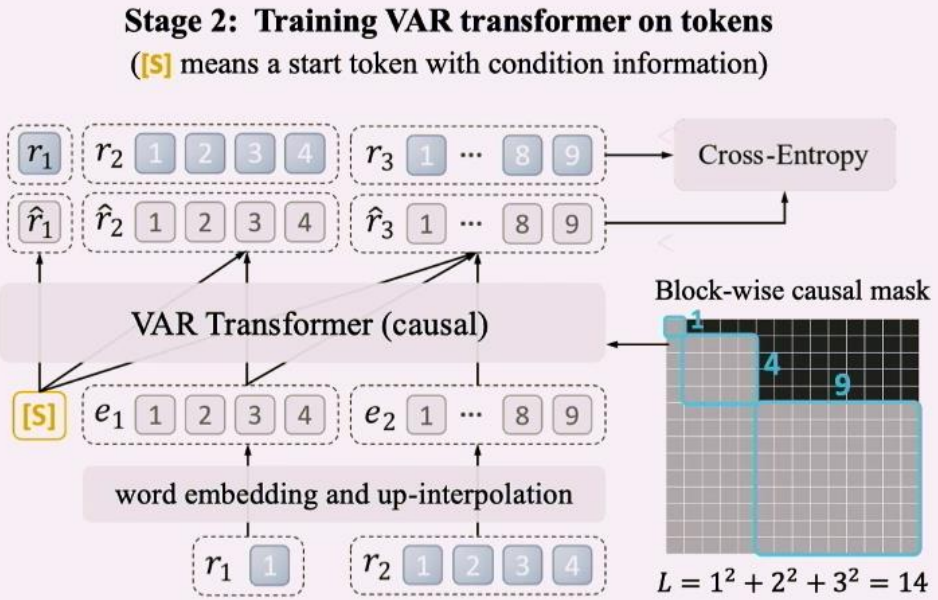
Inputs: raw image im;
Hyperparameters: steps K, resolutions $(h_k, w_k)_{k=1}^{K}$;
$f = \mathcal{E}(im)$, $R = []$;
for $k = 1, \cdots, K$ **do**
  $r_k = \mathcal{Q}(\text{interpolate}(f, h_k, w_k))$;
  $R = \text{queue\_push}(R, r_k)$;
  $z_k = \text{lookup}(Z, r_k)$;
  $z_k = \text{interpolate}(z_k, h_K, w_K)$;
  $f = f - \phi_k(z_k)$;
Return: multi-scale tokens R;

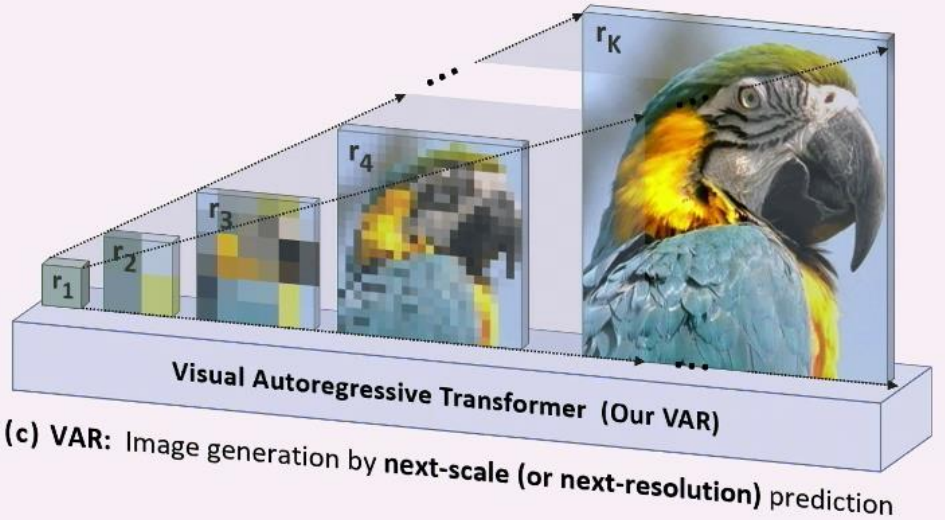
Algorithm 2: Multi-scale VQVAE Reconstruction

Inputs: multi-scale token maps R;
Hyperparameters: steps K, resolutions $(h_k, w_k)_{k=1}^{K}$;
$\hat{f} = 0$;
for $k = 1, \cdots, K$ **do**
  $r_k = \text{queue\_pop}(R)$;
  $z_k = \text{lookup}(Z, r_k)$;
  $z_k = \text{interpolate}(z_k, h_K, w_K)$;
  $\hat{f} = \hat{f} + \phi_k(z_k)$;
$\hat{im} = \mathcal{D}(\hat{f})$;
Return: reconstructed image $\hat{im}$;

- a new multi-scale quantization autoencoder to encode an image to K multi-scale discrete token maps $R = (r_1, r_2, \ldots, r_K)$ necessary for VAR learning

Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction





(c) **VAR:** Image generation by **next-scale (or next-resolution)** prediction

- The first step in the autoregressive process is to use the starting token [S] to predict the initial 1x1 token map.
- Subsequently, at each step, VAR **predicts the next larger-scale token map based on all previous token maps**(casual).
- During training, VAR uses standard cross-entropy loss to supervise the probability prediction of these token maps.
- During testing, the sampled token maps are converted to continuous embedding, and decoded using the VQVAE decoder to produce the final generated image.

Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction

Type	Model	FID↓	IS↑	Pre↑	Rec↑	#Para	#Step	Time
GAN	BigGAN [13]	6.95	224.5	**0.89**	0.38	112M	1	–
GAN	GigaGAN [43]	3.45	225.5	0.84	**0.61**	569M	1	–
GAN	StyleGan-XL [75]	2.30	265.1	0.78	0.53	166M	1	0.3 [75]
Diff.	ADM [26]	10.94	101.0	0.69	0.63	554M	250	168 [75]
Diff.	CDM [37]	4.88	158.7	–	–	–	8100	–
Diff.	LDM-4-G [71]	3.60	247.7	–	–	400M	250	–
Diff.	DiT-L/2 [64]	5.02	167.2	0.75	0.57	458M	250	31
Diff.	DiT-XL/2 [64]	2.27	278.2	0.83	0.57	675M	250	45
Diff.	L-DiT-3B [3]	2.10	304.4	0.82	0.60	3.0B	250	>45
Diff.	L-DiT-7B [3]	2.28	316.2	0.83	0.58	7.0B	250	>45
Mask.	MaskGIT [17]	6.18	182.1	0.80	0.51	227M	8	0.5 [17]
Mask.	RCG (cond.) [52]	3.49	215.5	–	–	502M	20	1.9 [52]
AR	VQVAE-2† [69]	31.11	∼45	0.36	0.57	13.5B	5120	–
AR	VQGAN† [30]	18.65	80.4	0.78	0.26	227M	256	19 [17]
AR	VQGAN [30]	15.78	74.3	–	–	1.4B	256	24
AR	VQGAN-re [30]	5.20	280.3	–	–	1.4B	256	24
AR	ViTVQ [92]	4.17	175.1	–	–	1.7B	1024	>24
AR	ViTVQ-re [92]	3.04	227.4	–	–	1.7B	1024	>24
AR	RQTran. [51]	7.55	134.0	–	–	3.8B	68	21
VAR	VAR-$d16$	3.30	274.4	0.84	0.51	310M	10	0.4
VAR	VAR-$d20$	2.57	302.6	0.83	0.56	600M	10	0.5
VAR	VAR-$d24$	2.09	312.9	0.82	0.59	1.0B	10	0.6
VAR	VAR-$d30$	1.92	323.1	0.82	0.59	2.0B	10	1
VAR	VAR-$d30$-re	**1.73**	**350.2**	0.82	0.60	2.0B	10	1
	(validation data)	1.78	236.9	0.75	0.67			

ImageNet 256×256 conditional generation

Type	Model	FID↓	IS↑	Time
GAN	BigGAN [13]	8.43	177.9	–
Diff.	ADM [26]	23.24	101.0	–
Diff.	DiT-XL/2 [64]	3.04	240.8	81
Mask.	MaskGIT [17]	7.32	156.0	0.5†
AR	VQGAN [30]	26.52	66.8	25†
VAR	VAR-$d36$-s	**2.63**	**303.2**	1

ImageNet 512×512 conditional generation

- **Sota performance on imagenet benchmark.**
- **Super fast inference speed**
- **VAR significantly advances traditional AR capabilities.**
- **Better perfomance than DiT(sora base model)**

To our knowledge, this is the first time of autoregressive models outperforming Diffusion transformers!

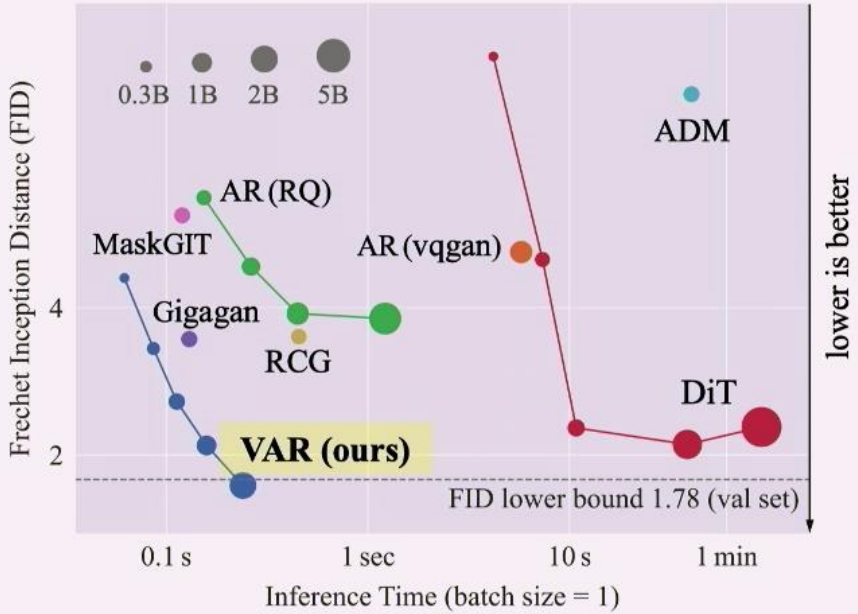
Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction

	Description	Para.	Model	AdaLN	Top-k	CFG	Cost	FID↓	Δ
1	AR [30]	227M	AR	✗	✗	✗	1	18.65	0.00
2	AR to VAR	207M	VAR-$d16$	✗	✗	✗	0.013	5.22	−13.43
3	+AdaLN	310M	VAR-$d16$	✓	✗	✗	0.016	4.95	−13.70
4	+Top-k	310M	VAR-$d16$	✓	900	✗	0.016	4.64	−14.01
5	+CFG	310M	VAR-$d16$	✓	900	1.5	0.022	3.60	−15.05
5	+Attn. Norm.	310M	VAR-$d16$	✓	900	1.5	0.022	3.30	−15.35
6	+Scale up	2.0B	VAR-$d30$	✓	900	1.5	0.052	1.73	−16.85

VAR significantly improve AR baseline :

- **improving Fréchet inception distance (FID) from 18.65 to 1.73**
- **inception score (IS) from 80.4 to 350.2**
- **with 20× faster inference speed**

Visual Autoregressive Modeling: scalability & scaling law

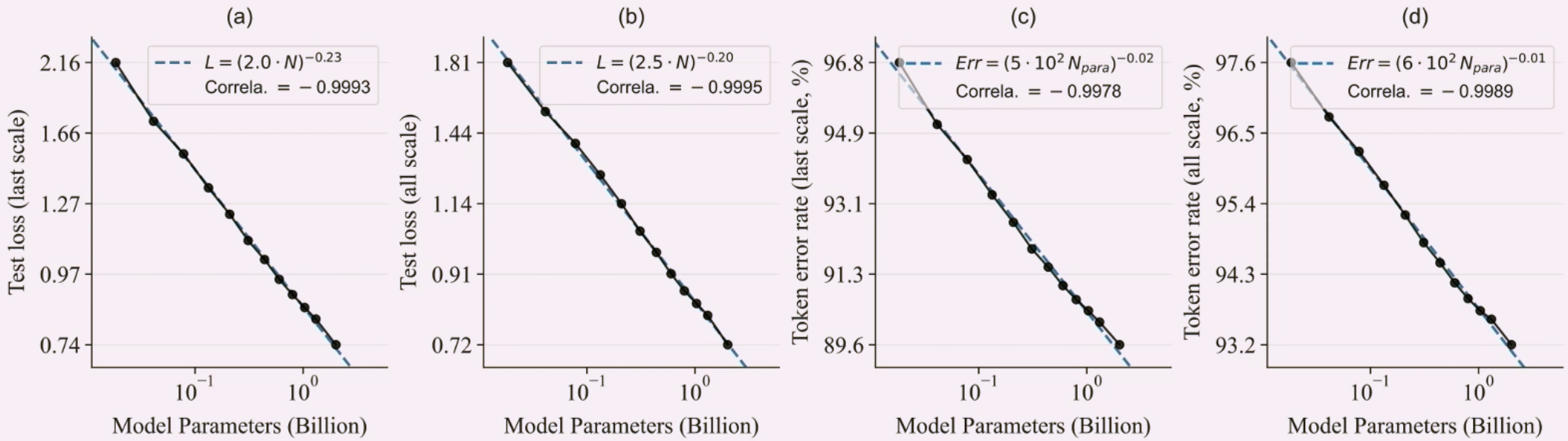


Type	Model	FID↓	IS↑	Pre↑	Rec↑	#Para	#Step	Time
Diff.	DiT-L/2 [63]	5.02	167.2	0.75	0.57	458M	250	31
Diff.	DiT-XL/2 [63]	2.27	278.2	0.83	0.57	675M	250	45
Diff.	L-DiT-3B [3]	2.10	304.4	0.82	0.60	3.0B	250	>45
Diff.	L-DiT-7B [3]	2.28	316.2	0.83	0.58	7.0B	250	>45
VAR	VAR-*d*16	3.30	274.4	0.84	0.51	310M	10	0.4
VAR	VAR-*d*20	2.57	302.6	0.83	0.56	600M	10	0.5
VAR	VAR-*d*24	2.09	312.9	0.82	0.59	1.0B	10	0.6
VAR	VAR-*d*30	1.92	323.1	0.82	0.59	2.0B	10	1
VAR	VAR-*d*30-re	**1.73**	**350.2**	0.82	0.60	2.0B	10	1
	(validation data)	1.78	236.9	0.75	0.67			

Compared with Diffusion Transformer (DiT, base model for sora, stable diffusion3), VAR demonstrates:

- **Better performance**: After scaling up, VAR achieves a FID of 1.80, significantly outperforming DiT's best score of 2.10.
- **Faster speed:** VAR requires less than 0.3 seconds to generate a 256x256 image, making it 45 times faster than DiT; for 512x512 images, it is 81 times faster.
- **Better scaling capability:** DiT's large-scale models exhibit saturation when expanded to 3B and 7B parameters, failing to approach the FID lower bound. In contrast, VAR continues to improve as it scales up to 2 billion parameters, eventually reaching the FID lower bound.
- **More efficient data utilization:** VAR surpasses DiT with just 350 epochs of training, compared to DiT's 1400 epochs of training.

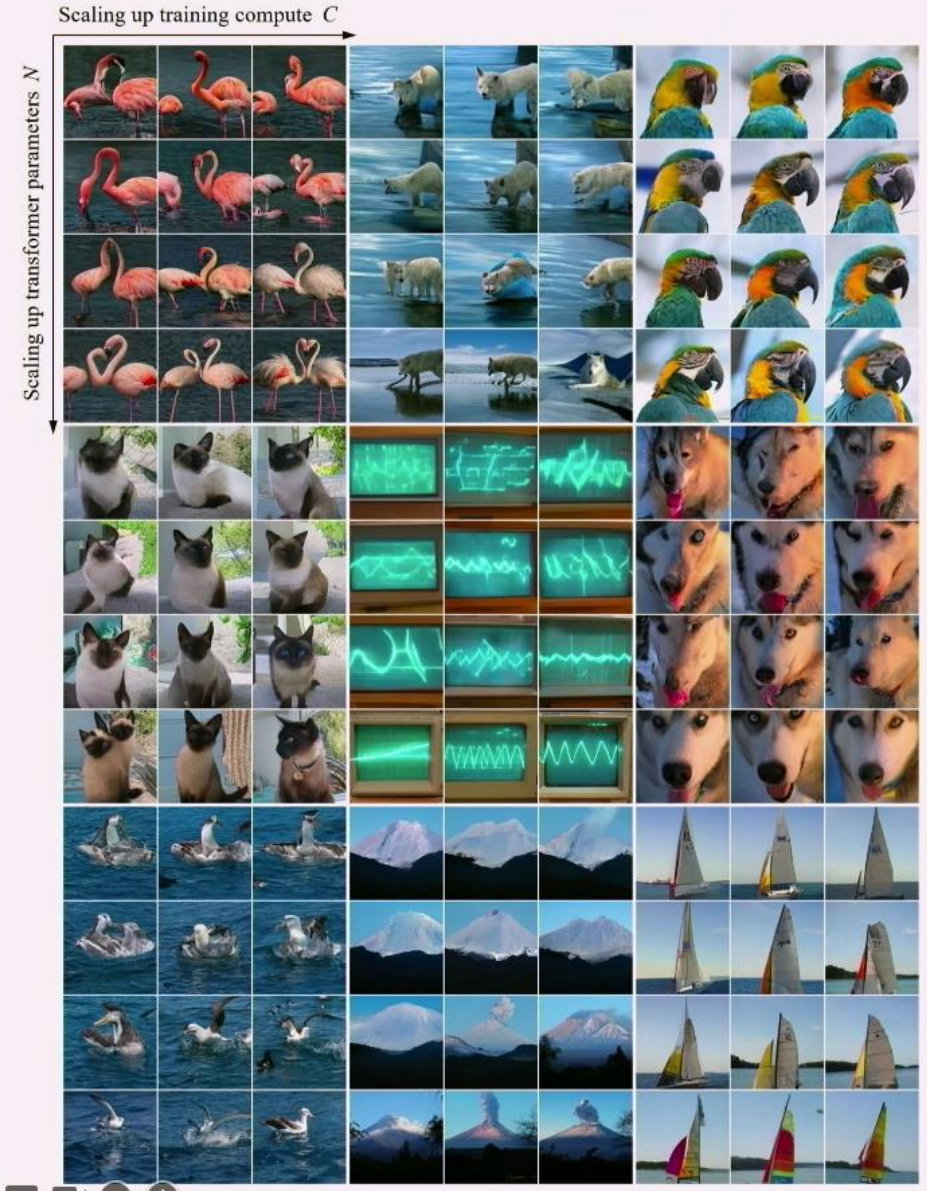
Visual Autoregressive Modeling: scalability & scaling law

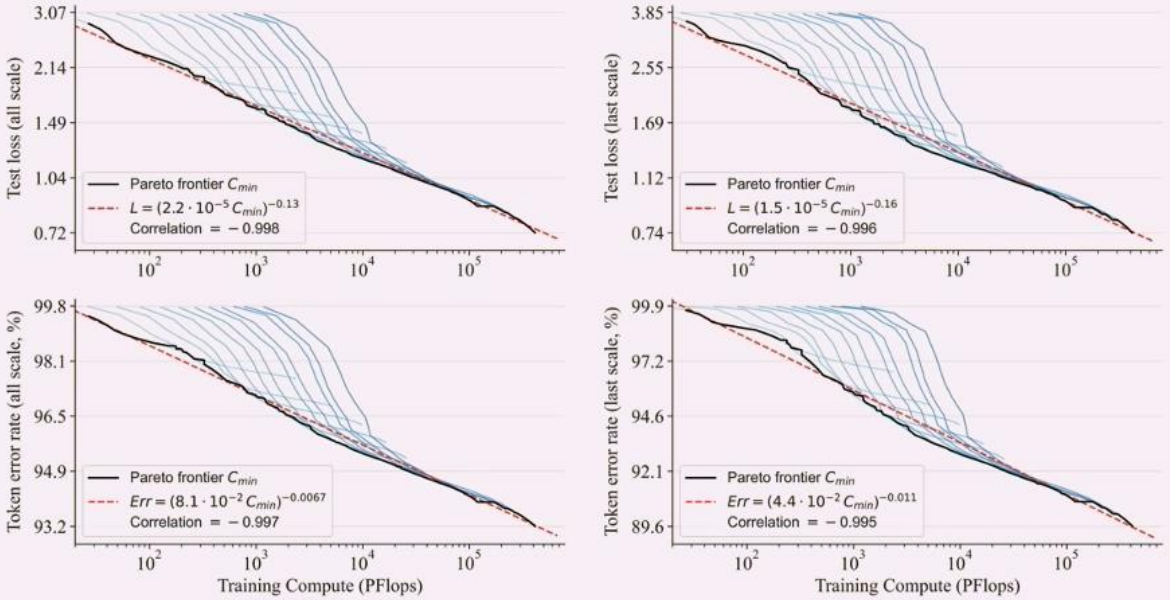


VAR demonstrates an almost perfectly consistent power-law Scaling Law with LLM:

- **The token accuracy or cross-entropy on the validation set predictably decreases as the model size increases.**
- **Exhibits a power-law relationship, similar to LLMs, showing a linear relationship when log-scaled, with a correlation coefficient of -0.998.**

Visual Autoregressive Modeling: scalability & scaling law

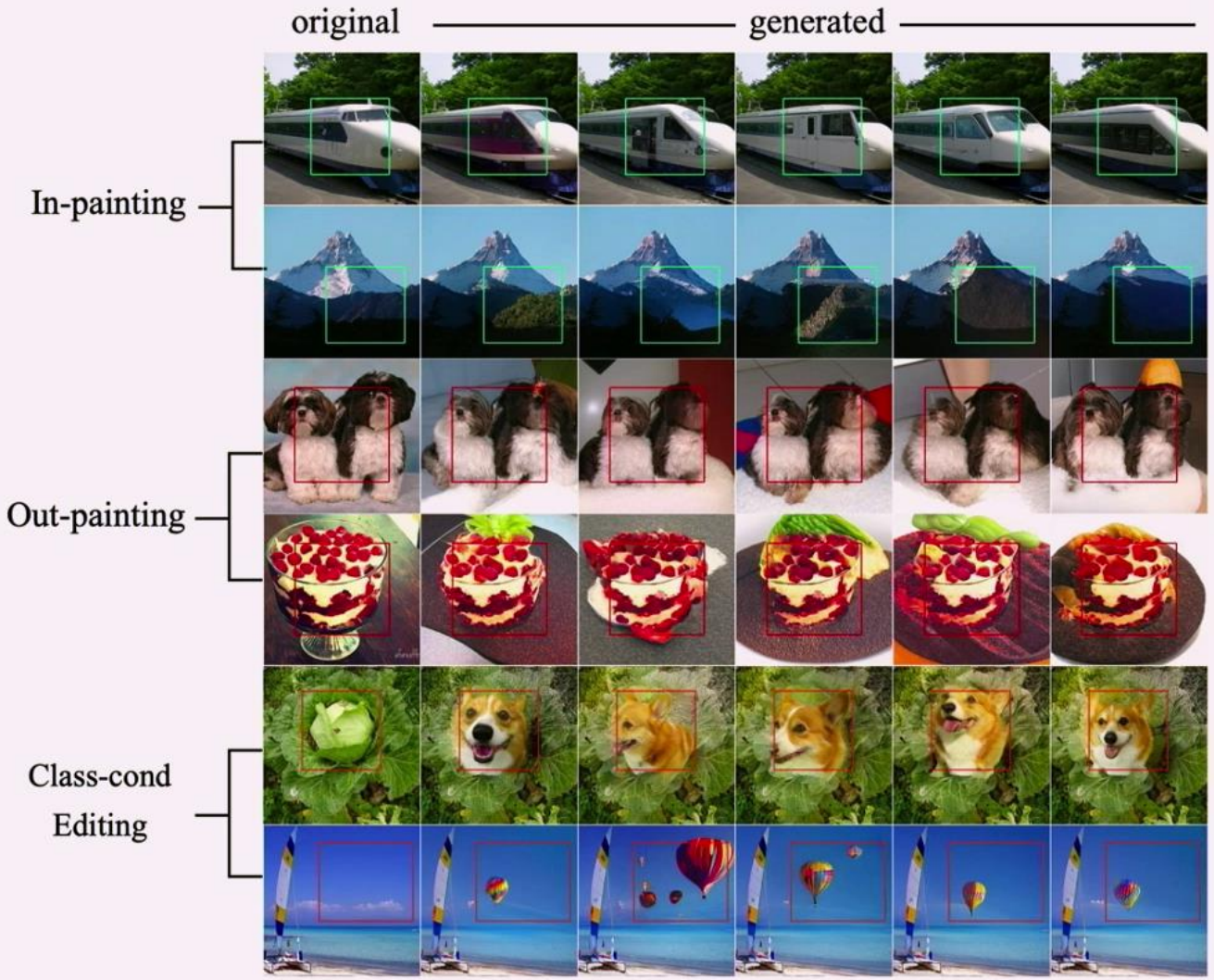




Scaling laws with optimal training compute C_{min}

- **Scaling up the model parameters leads to a gradual and noticeable improvement in the model's generative capabilities.**
- **Scaling up the model's computation also leads to a gradual and noticeable improvement in its generative capabilities**

Visual Autoregressive Modeling: Scalability & Zero-shot Generalization

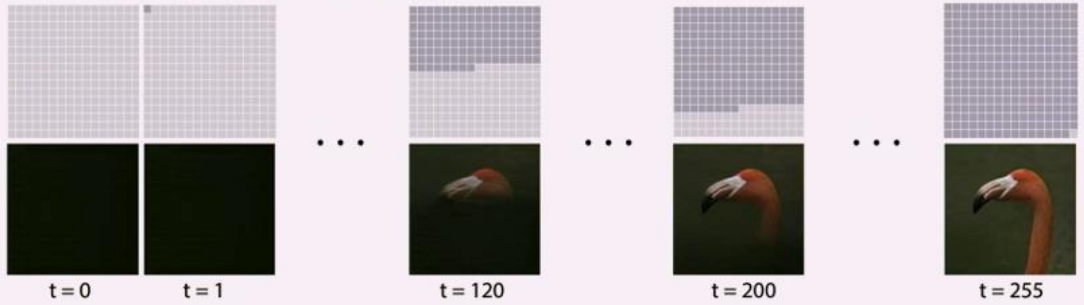


Zero-shot generalization

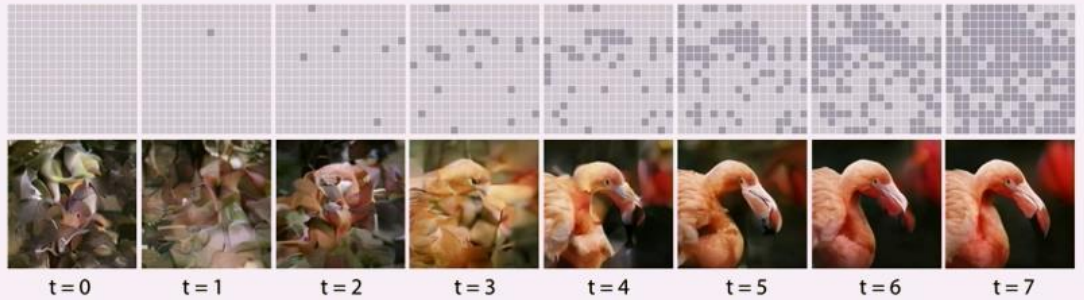
- **A well-trained VAR Transformer on conditional generation tasks can zero-shot generalize to various generative tasks without any fine-tuning, achieving impressive results.**
  - **Image in-painting**
  - **Image out-painting**
  - **Image class-condition editing**

“Visual” GPT lives in scale-space!

Parti’s autoregressive order



MUSE’s autoregressive order



VAR:

1 token
16x16 pix

autoregressive in the scale space

16x16 tokens
256x256 pix

- **faster**: Parti $O(L^3)$, MUSE $O(TL^2)$, VAR $O(2L^2)$
- **more natural**: more similar to human’s visual perception process
- **more reasonable** order:
  - VQ-VAE’s token is not NLP word.
  - VQ-VAE’s token map is an output of global attention; there’s no independency at all!

Visual Autoregressive Modeling

Highlight

- A new visual generative framework using a multi-scale autoregressive paradigm with **next-scale prediction**, offering new insights in autoregressive algorithm design for computer vision.
- An empirical validation of **VAR models' Scaling Laws** and zero-shot generalization potential, which initially emulates the appealing properties of large language models (LLMs).
- A breakthrough in visual autoregressive model performance, making GPT-style autoregressive methods surpass strong diffusion models in image synthesis **for the first time**.

Visual AutoRegressive in the scale space

1 token
16x16 pix

16x16 tokens
256x256 pix

[A] Keyu Tian, Yi Jiang et al., Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction, 2024 Arxiv.

Questions from you

- I am wondering how sensitive is generation quality to the particular choice of scale sequence, and is it possible to have different scales within the entire autoregressive process.
- Is VAR still considered an influential approach to image generation in 2026, or have newer methods largely replaced it?
- I would be interested to know how exactly they would go about extending this work to cover video generation and what the most optimal way to go about traversing the 3D space might be.

VAR extensions

- **2025-11:** We Release our Text-to-Video generation model **InfinityStar** based on VAR & Infinity, please check Infinity⭐.
- **2025-11:** 🎉 InfinityStar is accepted as **NeurIPS 2025 Oral.**
- **2025-04:** 🎉 Infinity is accepted as **CVPR 2025 Oral.**
- **2024-12:** 🏆 VAR received **NeurIPS 2024 Best Paper Award**.
- **2024-12:** 🔥 We Release our Text-to-Image research based on VAR, please check Infinity.
- **2024-09:** VAR is accepted as **NeurIPS 2024 Oral** Presentation.
- **2024-04:** Visual AutoRegressive modeling is released.

https://github.com/FoundationVision/VAR

The University
of North Carolina
at Chapel Hill