Broken Neural Scaling Laws (BNSL)

Ethan Caballero
Kshitij Gupta,
Irina Rish,
David Krueger

arxiv.org/abs/2210.14891

ICLR 2023 Conference Paper


AGI

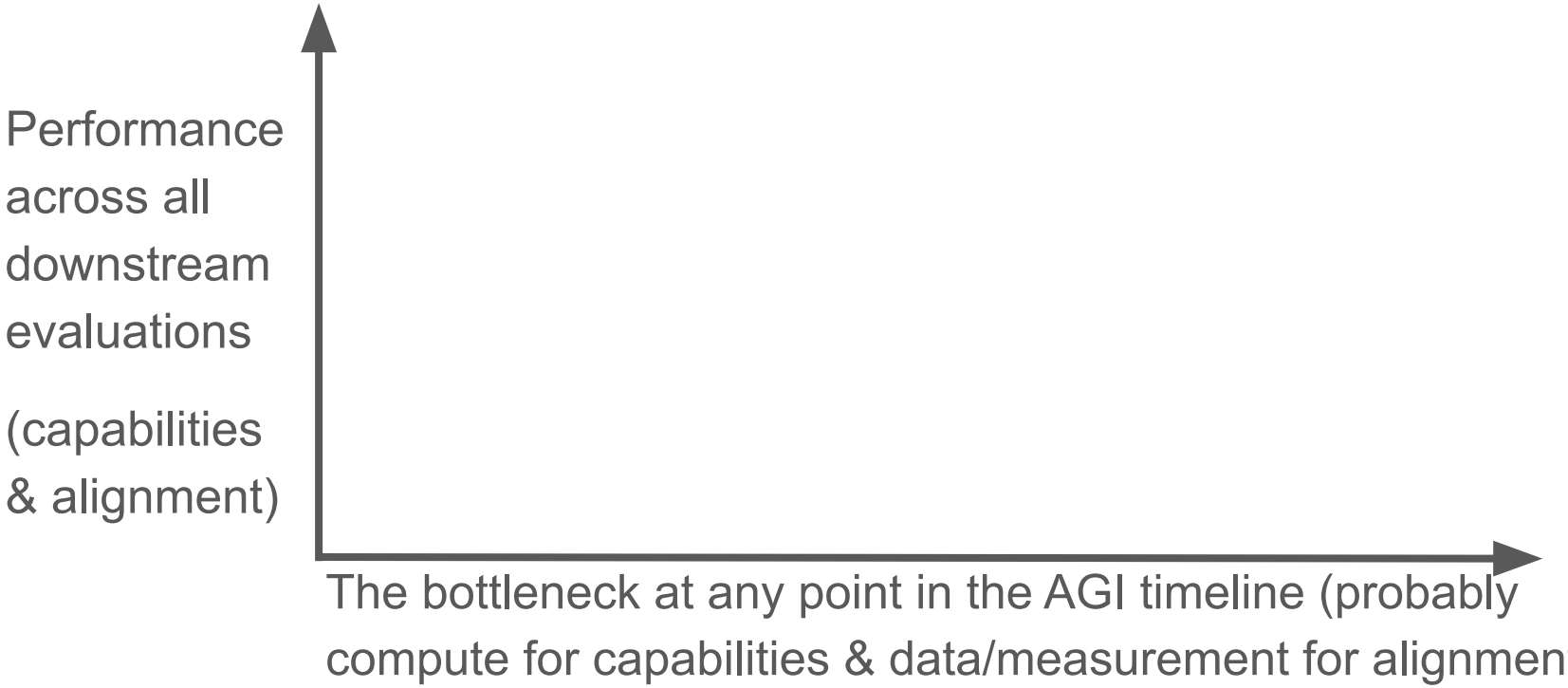
AGI = highly autonomous systems that outperform humans at most economically valuable work

There are multiple paths to AGI.

However, the only path that matters is the fastest path, because economic/military competition is selecting for it and as a result it’s the path that is/will_be deployed in the real world.

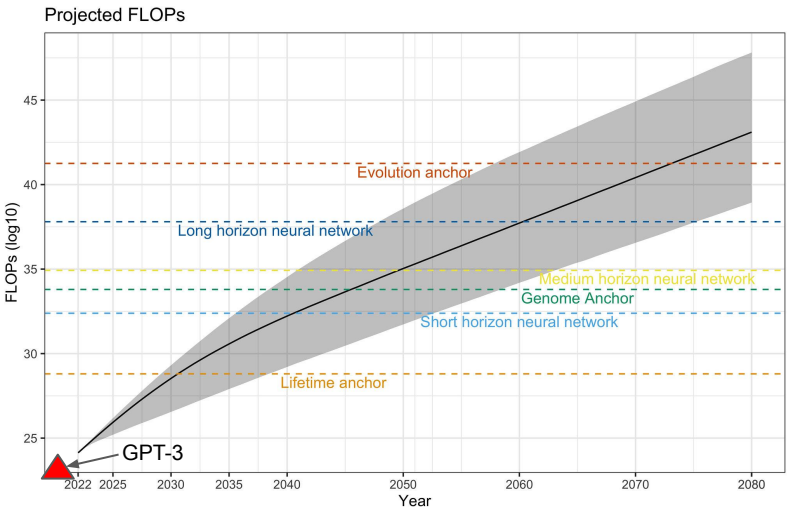
If one wants to steer AGI to be maximally beneficial, identifying the fastest path as early as possible is critical.

The fastest path to maximally beneficial AGI is that which scales best according to all downstream evaluations (that matter) ~simultaneously

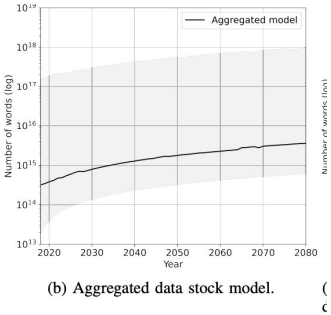


The Bottleneck(s) being scaled (i.e. the x-axis)

Compute

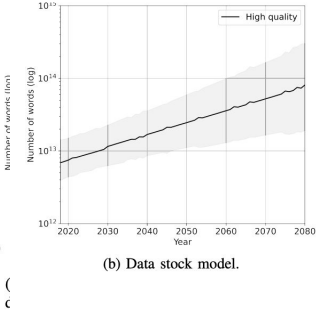


Data



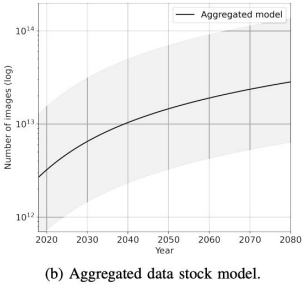
(b) Aggregated data stock model.

ig. 3: Models of low-quality language data



(b) Data stock model.

. 4: Models of high-quality language data



(b) Aggregated data stock model.

Fig. 5: Models of vision data.

The x-axes (i.e. quantities being scaled) we test in the paper are **amount of compute used for training, number of model parameters, training dataset size, model input size, retrieval dataset size, number of submodels within a larger model, number of training steps, and upstream performance**.

The Performance Evaluation Metric(s) (i.e. the y-axis)

prediction error,

cross entropy,

calibration error,

AUROC,

BLEU score percentage,

F1 score,

reward,

Elo rating,

FID score

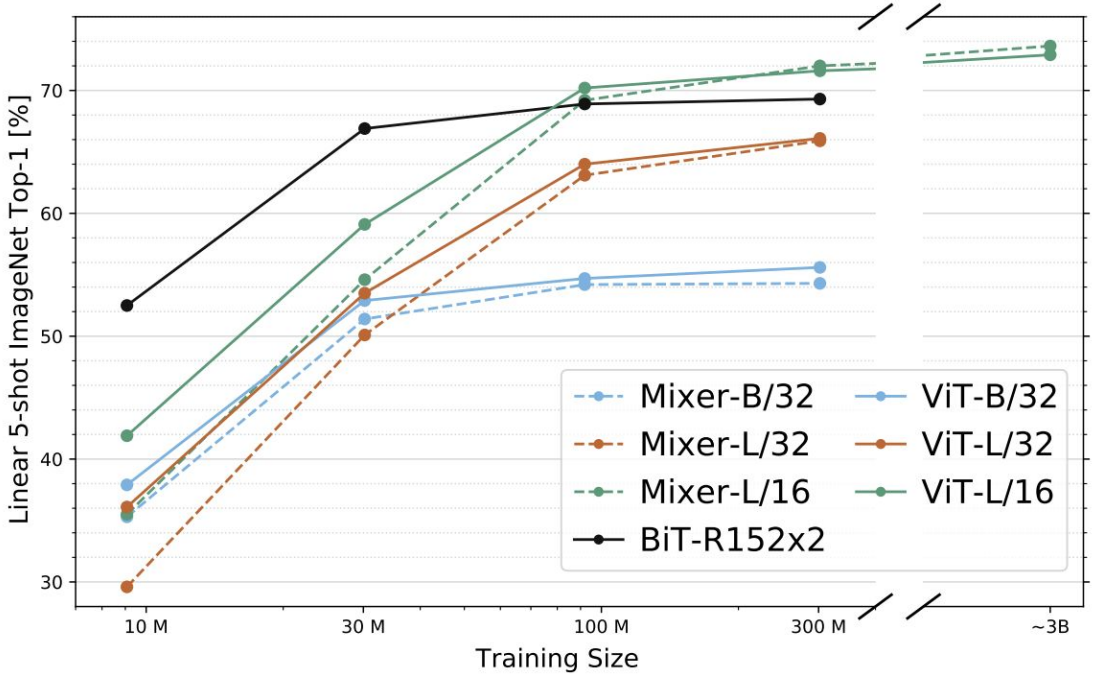
~Anything??

(downstream and/or upstream)

(0-Shot & N-Shot, and/or Finetuned)

(Prompted (& not Prompted))

Methods that perform best at smaller scales often are not the best performing at larger scales

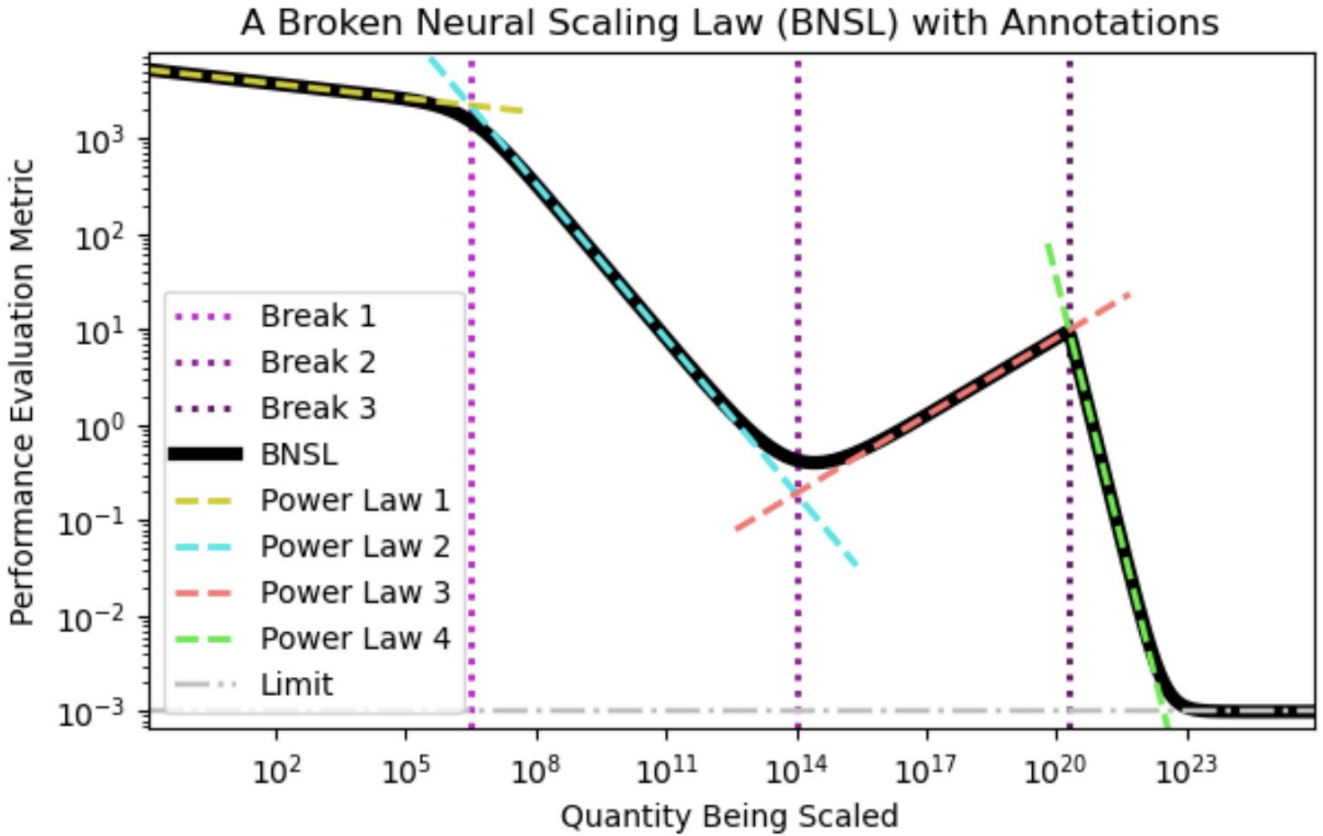


What characterizes the relationship between the x-axis (quantity being scaled) and the y-axis (performance evaluation metric) such that we can extrapolate/predict what the value of y will be for any given (i.e. larger) value of x?

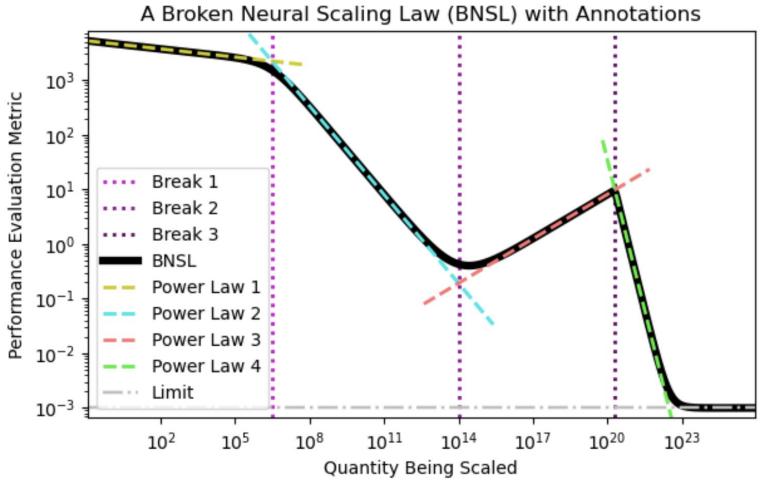
Broken Neural Scaling Law (BNSL):

$$y = a + \left(bx^{-c_0}\right) \prod_{i=1}^{n} \left(1 + \left(\frac{x}{d_i}\right)^{1/f_i}\right)^{-c_i * f_i}$$

Dark Black Solid line is a Broken Neural Scaling Law (BNSL):



$$y = a + \left(bx^{-c_0}\right) \prod_{i=1}^{n} \left(1 + \left(\frac{x}{d_i}\right)^{1/f_i}\right)^{-c_i * f_i}$$



x = quantity being scaled ; **y** = performance evaluation metric

n = number of (smooth) “breaks” (i.e. transitions) between n + 1 consecutive approximately linear (on a log-log plot) segments, for a total of n + 1 approximately linear segments (on a log-log plot).

a = the limit as to how far the value of y (performance evaluation metric) can be reduced (or maximized) even if x (the quantity being scaled) goes to infinity.

b = the offset of functional form on a log-log plot (analogous to the intercept b in y = mx + b on a linear-linear plot).

c_0 = represents the slope of the first approximately linear region on a log-log plot.

c_i = the difference in slope of the (i)th approximately linear region and (i + 1)th approximately linear region on a log-log plot.

d_i = where on the x-axis the break between the (i)th and the (i+ 1)th approximately linear region (on a log-log plot) occurs.

f_i = the sharpness of break between the (i)th and the (i + 1)th approximately linear region on a log-log plot; smaller (nonnegative) values of f_i yield a sharper break and intervals (before and after the (i)th break) that are more linear on a log-log plot; larger values of f_i yield a smoother break and intervals (before and after the (i)th break) that are less linear on a log-log plot.

Results; All the scaling behaviors that we show BNSL accurately models and extrapolates:

Downstream and Upstream ; Zero-Shot, Prompted, and Fine-Tuned

Large-Scale Vision, Language, Audio, Video, Diffusion, Generative Modeling, Multimodal Learning, Contrastive Learning, AI Alignment, Robotics, Out-Of-Distribution Generalization, Continual Learning, Transfer Learning, Uncertainty Estimation / Calibration, Out-Of-Distribution Detection, Adversarial Robustness, Distillation, Sparsity, Retrieval, Quantization, Pruning, Fairness, Molecules, Computer Programming/Coding, Math Word Problems, Arithmetic, Double Descent, “Emergent” “Phase Transitions”, Unsupervised / Self-Supervised Learning, & Reinforcement Learning (Single Agent & Multi-Agent)

ResNet, Transformer, MLP-Mixer, Graph Neural Network, U-Net, Sparsely-Gated Mixture-of-Experts, Sparse Pruned Model

X-axes: Compute, Dataset Size, Number of Training Steps, Input (e.g. Context) Size, Number of Model Parameters, & Upstream Performance

Y-axes: prediction error, cross entropy, calibration error, AUROC, BLEU score percentage, F1 score, reward, Elo rating, FID score

Extrapolating downstream scaling behavior of Large Scale Vision and Language

Various Architectures:

(BiT) Big Residual Networks, MLP mixers (MiX), and vision transformers (ViT), language transformer (decoder only) (encoder-decoder)

Pretrain on subsets of JFT-300M (300 Million images), and Pretrain on subsets of large amounts of text from the internet (hundreds of billions of tokens).

Evaluated on each of a large number of downstream task evaluations.

Functional forms from previous work that we compare to

M1 $$y = ax^b$$

M2 $$y = ax^b + c$$

M3 $$y = a(x + d)^b + c$$

M4 $$(y - \epsilon_\infty)/((\epsilon_0 - y)^a) = bx^c$$

Domain	M1 ↑	M2 ↑	M3 ↑	M4 ↑	BNSL ↑
Downstream Image Classification	2.78%	4.17%	9.72%	13.89%	**69.44%**
Language (Downstream and Upstream)	10%	5%	10%	0%	**75%**

Table 2: Percentage of tasks by domain where each functional form is the best for extrapolation of scaling behavior. Numbers for M1, M2, M3, and M4 were obtained via correspondence with authors of Alabdulmohsin et al. (2022). See Sections 5.1 and 5.2 for more details.

- Percentage of tasks by domain where each functional form is the best for extrapolation of scaling behavior.

Plots

You are about to see a bunch of plots.

In all plots, **black** points are the points used for fitting a BNSL, **green** points are the held-out points used for evaluating extrapolation of BNSL fit to the black points, and a **red** line is the BNSL that has been fit to the black points. 100% of the plots contain **green** point(s) for evaluating extrapolation.

Except when stated otherwise, the interval of the plot contains one break of BNSL fit to black points.

BNSL accurately extrapolates to scales that are **over an order of magnitude** away (even downstream)
(e.g. with (pre-)training dataset size on the x-axis)
(e.g. for Vision Tasks)

A.7 EXTRAPOLATION OF DOWNSTREAM PERFORMANCE TO SCALES THAT ARE AN OVER ORDER OF MAGNITUDE LARGER THAN THE MAXIMUM (ALONG THE X-AXIS) OF THE POINTS USED FOR FITTING

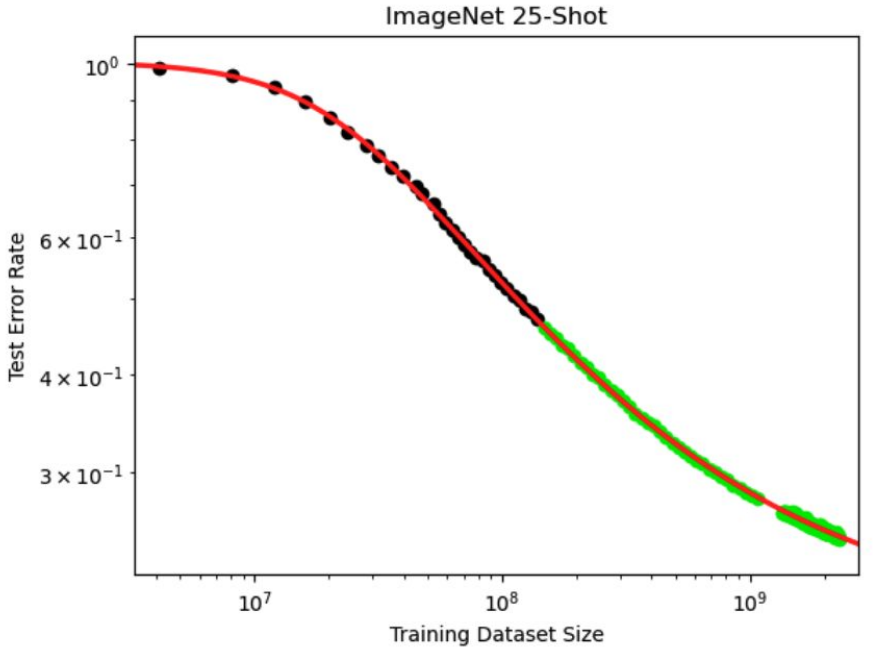


Figure 6: Extrapolation Results of BNSL for Downstream Performance to Scales that are an Order of Magnitude larger than the maximum (along the x-axis) of the points used for fitting. Experimental data of scaling behavior obtained from scaling laws benchmark of Alabdulmohsin et al. (2022). The upstream task is supervised pretraining of MLP mixers (MiX) (Tolstikhin et al., 2021) on subsets (i.e. the x-axis of plot) of JFT-300M (Sun et al., 2017). The Downstream Task is n-shot ImageNet classification (i.e. the y-axis of plot). See Section A.7 for more details.

BNSL accurately extrapolates the scaling behavior of **Downstream Language Tasks** to scales that are over **an orders of magnitude** away (e.g. with number of model parameters on the x-axis)

A.14 EXTRAPOLATION RESULTS FOR DOWNSTREAM LANGUAGE TASKS (WHEN NUMBER OF MODEL PARAMETERS IS ON THE X-AXIS) TO SCALES THAT ARE OVER AN ORDER OF MAGNITUDE LARGER THAN THE MAXIMUM (ALONG THE X-AXIS) OF THE POINTS USED FOR FITTING.

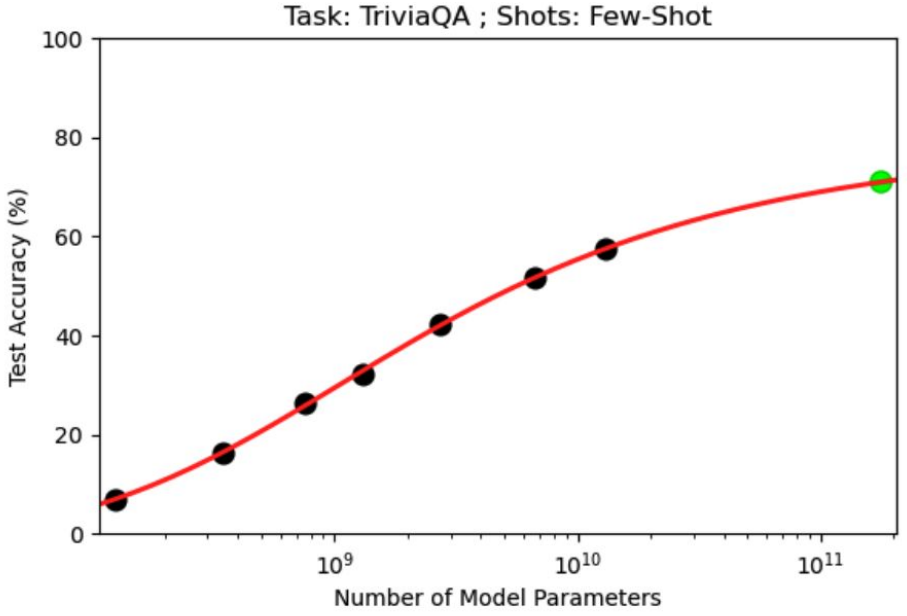


Figure 13: Extrapolation Results of BNSL for Downstream Language Tasks when Number of Model Parameters is on the x-axis. "Few-Shot" in plot title means few-shot prompting is used for that downstream evaluation as described in GPT-3 arXiv paper (Brown et al., 2020). Experimental data obtained from Table H.1 of the GPT-3 arXiv paper (Brown et al., 2020). See Section A.14 for more details.

BNSL accurately extrapolates scaling behavior of **Downstream Performance** to scales that are **over 100,000 times larger** than the maximum (along the x-axis) of the points used for fitting (e.g. with amount of compute used for training on the x-axis) (e.g. for language/coding)

A.32 EXTRAPOLATION RESULTS FOR DOWNSTREAM PERFORMANCE (E.G. ON LANGUAGE/CODING TASKS) TO SCALES THAT ARE **GREATER THAN 100,000 TIMES LARGER** THAN THE MAXIMUM (ALONG THE X-AXIS (E.G. AMOUNT OF COMPUTE USED FOR TRAINING)) OF THE POINTS USED FOR FITTING

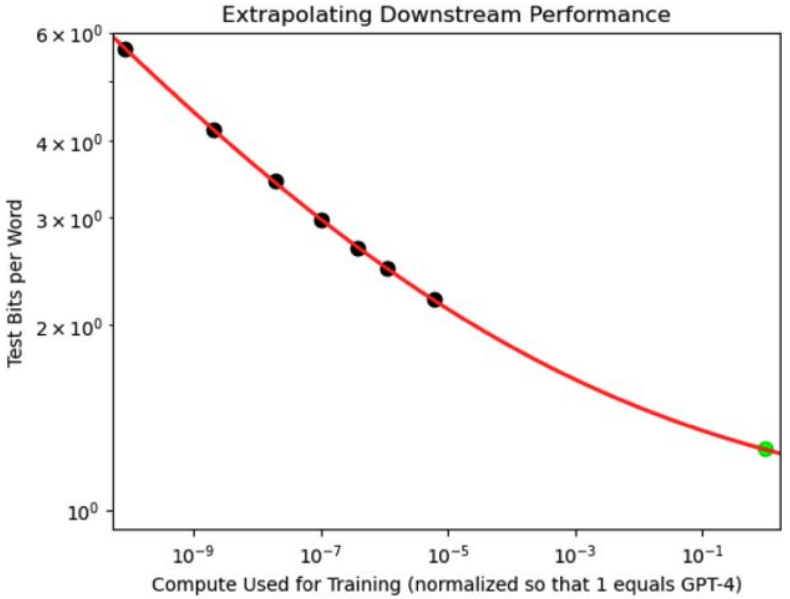


Figure 31: Extrapolation Results of BNSL for Downstream Performance (e.g. on Language/Coding Tasks) to scales that are **greater than 100,000 (i.e. greater than 5 orders of magnitude) times larger** than the maximum (along the x-axis (e.g. amount of compute used for training)) of the points used for fitting. Experimental data obtained from Figure 1 of GPT-4 paper (OpenAI, 2023). Each point corresponds to a different individual training run. The downstream evaluation (i.e. the y-axis) in the plot is downstream zero-shot test bits per word on OpenAI's internal codebase (which was held out from the training set). The x-axis is normalized (i.e. divided by the amount of compute used to train GPT-4) such that 1 (i.e. where the green point is) is the amount of compute used to train GPT-4. Given that other language models released before GPT-4 used greater than 1e24 FLOPs of training compute and that GPT-4 is considerably more capable than those other language models, one can speculatively estimate that amount of compute used to train GPT-4 is probably an amount that is greater than 1e24 FLOPs. See Section A.32 for more details.

BNSL accurately extrapolates the scaling behavior of **Reinforcement Learning (Single-Agent and Multi-Agent)** (e.g. w/ compute, dataset size, or # of parameters on x-axis)

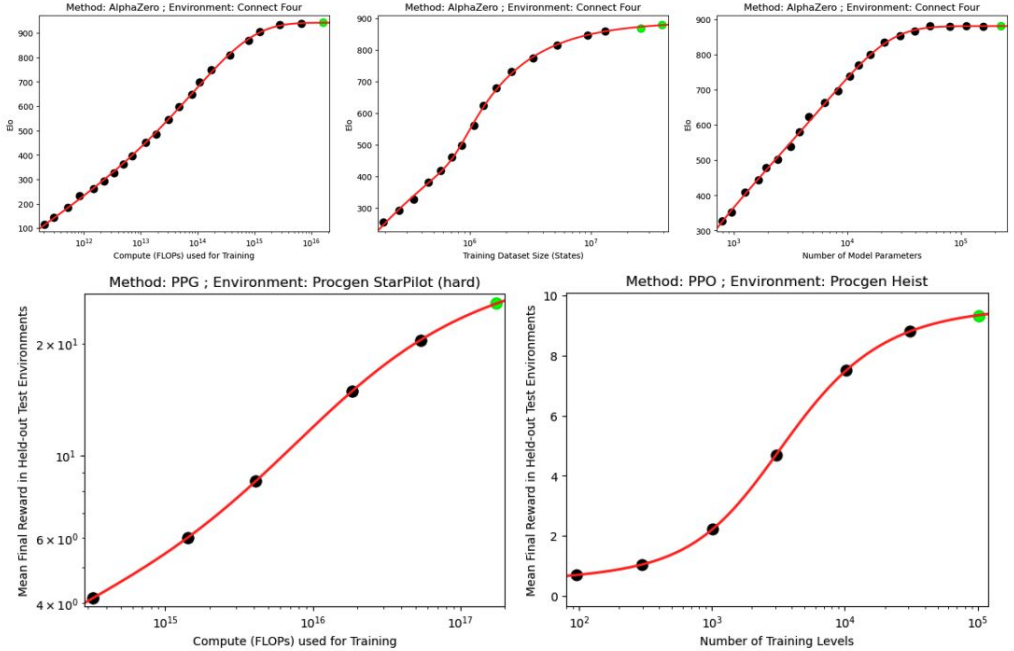


Figure 3: Extrapolation Results of BNSL on Reinforcement Learning Scaling Experimental Data. Experimental data of the top left plot and top middle plot and top right plot is from Figure 4 and Figure 5 and Figure 3 respectively of Neumann & Gros (2022). Experimental Data of the bottom left plot is from Figure 1 left of Hilton et al. (2023). Experimental Data of the bottom right plot is from Figure 2 of Cobbe et al. (2020). Top left and bottom left plot is the compute-optimal Pareto frontier. See Section 5.3 for more details.

BNSL accurately extrapolates the scaling behavior of **Diffusion Generative Models of Images**

A.10 Extrapolation Results for Diffusion Generative Models of Images

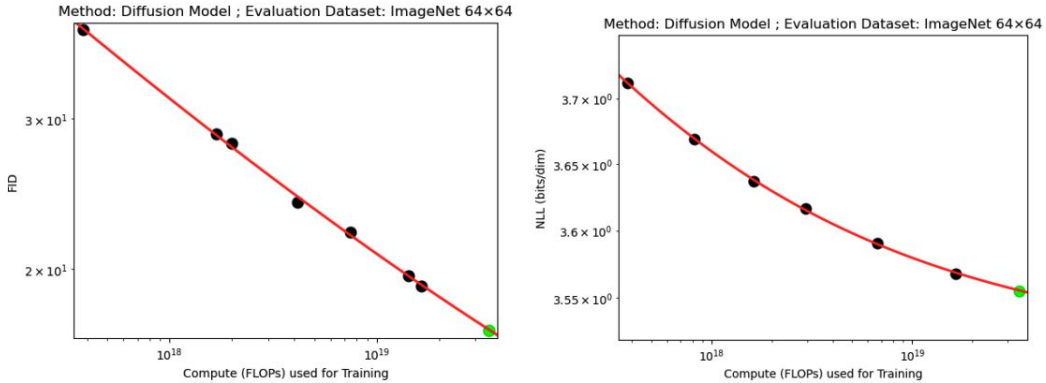


Figure 9: Extrapolation Results of BNSL for scaling behavior of Diffusion Generative Models of Images. Frechet Inception Distance (FID) score is on the y-axis in the left plot. Negative log-likelihood (NLL) is the y-axis in the right plot. For both plots, compute used for training is on the x-axis and Imagenet 64x64 is the evaluation dataset. Experimental data of scaling behavior obtained from Figure 10 of Nichol & Dhariwal (2021). See Section A.10 for more details.

In Figure 9, we show that BNSL accurately extrapolates the scaling behavior of Diffusion Generative Models of Images from Figure 10 of Nichol & Dhariwal (2021) when Negative Log-likelihood (NLL) or Frechet Inception Distance (FID) score is on the y-axis and compute used for training is on the x-axis; compute is scaled in the manner that is Pareto optimal with respect to the performance evaluation metric on the y-axis.

BNSL accurately extrapolates the scaling behavior of **Video**

A.11 EXTRAPOLATION RESULTS FOR GENERATIVE MODELS OF VIDEO

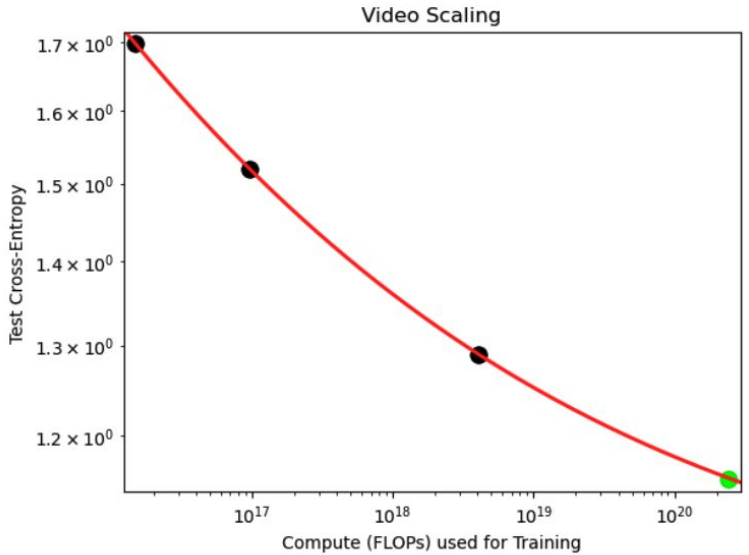


Figure 10: Extrapolation Results of BNSL for scaling behavior of Generative Models of Video. Upstream Test Cross-Entropy is on the y-axis. Videos scraped from the web are the evaluation dataset. During training, compute (used for training autoregressive transformer) on the x-axis is scaled in the manner that is Pareto optimal with respect to the performance evaluation metric on the y-axis. Experimental data of scaling behavior obtained from top right plot of Figure 5 of Henighan et al. (2020). See Section A.11 for more details.

In Figure 10, we show that BNSL accurately extrapolates the scaling behavior of generative models of video. Each frame is downsampled to a pixel resolution of 64x64; each frame is then tokenized via a pretrained 16x16 VQVAE (Van Den Oord et al., 2017) to obtain 256 tokens per frame. 16 consecutive frames are then input to an autoregressive transformer as a length 4096 (16x16x16) sequence. The dataset is 100 hours of videos scraped from the web. See section 2 of Henighan et al. (2020) for more details.

BNSL accurately extrapolates the scaling behavior of **AI Alignment (even downstream)**

A.21 EXTRAPOLATION RESULTS FOR AI ALIGNMENT VIA RLHF

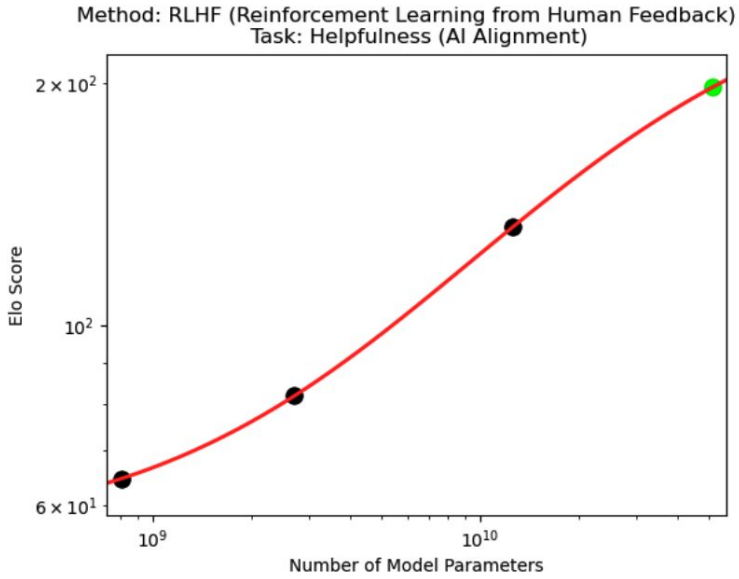


Figure 20: Extrapolation Results of BNSL for Downstream AI Alignment when Number of Model Parameters is on the x-axis. Experimental data obtained from the Static HH RLHF results from Figure 1 of Bai et al. (2022). See Section A.21 for more details.

In Figure 20, we find BNSL accurately extrapolates the scaling behavior of a pretrained language model finetuned (i.e. aligned) via Reinforcement Learning from Human Feedback (RLHF) to be helpful from Figure 1 of Bai et al. (2022). The y-axis is Elo score based on crowdworker preferences. The x-axis is the number of model parameters that the language model contains.

BNSL accurately extrapolates the scaling behavior of **Contrastive Learning & Multimodal Learning (even Downstream)**

A.30 EXTRAPOLATION RESULTS FOR DOWNSTREAM PERFORMANCE OF MULTIMODAL CONTRASTIVE LEARNING (I.E. NON-GENERATIVE UNSUPERVISED LEARNING)

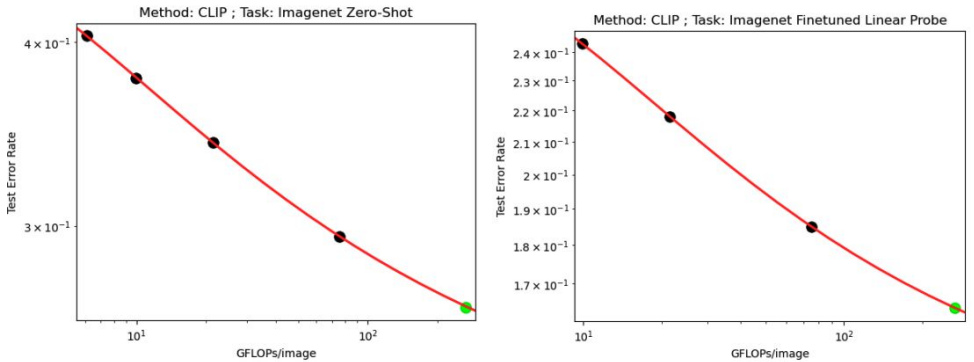


Figure 29: Extrapolation Results of BNSL for Downstream Performance of Multimodal Contrastive Learning (i.e. Non-Generative Unsupervised Learning). Experimental data of scaling behavior obtained from Table 10 and Table 11 in arXiv version of Radford et al. (2021). The upstream task is "Contrastive Image Language Pretraining" (a.k.a. CLIP) of ResNets on a training dataset consisting of hundreds of millions of image-text pairs. The x-axis is GFLOPs/image (GigaFLOPs/image) of the forward-pass of model. The Downstream Task is ImageNet classification (i.e. the y-axis of plot). The y-axis of left plot is Zero-Shot Downstream. The y-axis of right plot is performance of model with finetuned linear probe on it. See Section A.30 for more details.

In Figure 30, we show that BNSL accurately extrapolates the scaling behavior of the Downstream Performance of Multimodal Contrastive Learning (i.e. Non-Generative Unsupervised Learning).

BNSL accurately extrapolates the scaling behavior of **Continual Learning (i.e. Catastrophic Forgetting)**

A.22 Extrapolation Results for Continual Learning (i.e. Catastrophic Forgetting)

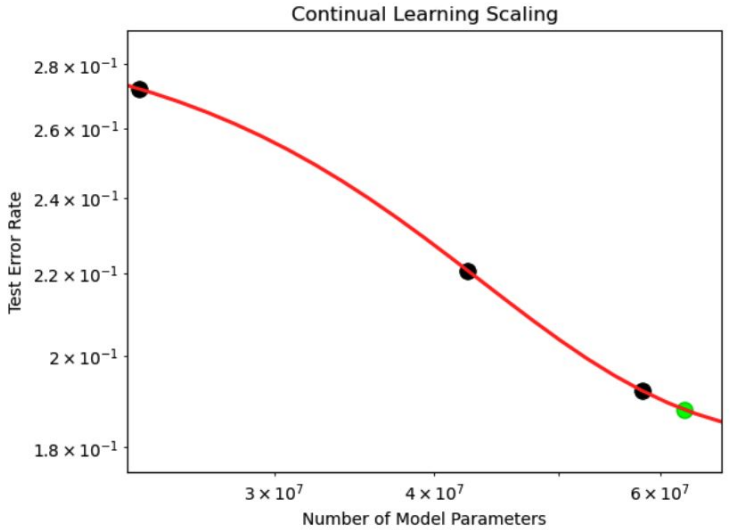


Figure 21: Extrapolation Results of BNSL for Continual Learning (i.e. Catastrophic Forgetting). Experimental data obtained from the Domainnet/Clipart section of the bottom right of Figure 2 of (Ramasesh et al., 2022). X-axis is number of model parameters in the ResNet model. In this setup, model is trained (in sequence, not simultaneously) on task A and then task B. Y-axis is mean of the test error rate on task A and task B. See Section A.22 for more details.

In Figure 21, we find that BNSL accurately extrapolates the scaling behavior of continual learning (i.e. catastrophic forgetting).

BNSL accurately extrapolates scaling behavior of **Robotics and Out-of-Distribution Generalization**

A.23 EXTRAPOLATION RESULTS FOR ROBOTICS (OUT-OF-DISTRIBUTION GENERALIZATION AND IN-DISTRIBUTION GENERALIZATION)

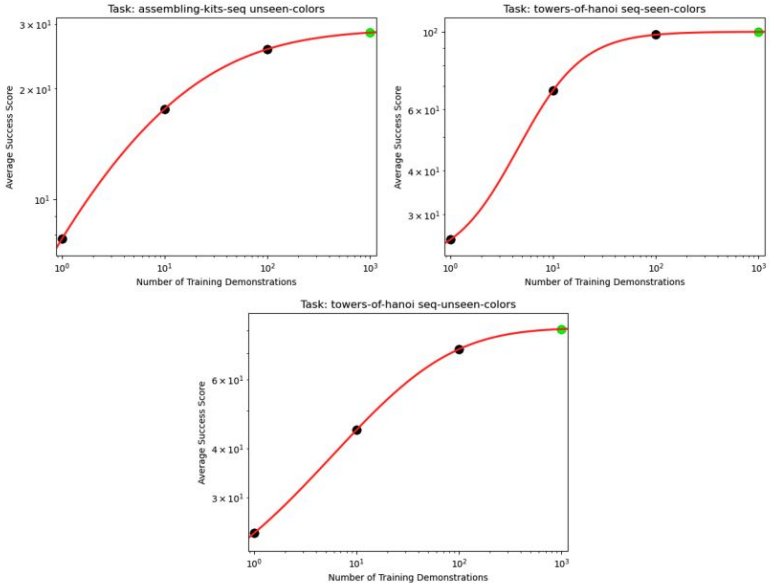


Figure 22: Extrapolation Results of BNSL for Robotic control (and Out-of-Distribution Generalization). Experimental data obtained from the transporter (Zeng et al., 2021) model results from Table 1 of Shridhar et al. (2021). X-axis is number of training demonstrations. Y-axis is task success score (mean percentage) obtained via 100 evaluations. See Section A.23 for more details.

In Figure 22, we find BNSL accurately extrapolates the scaling behavior of a transporter (Zeng et al., 2021) model trained via imitation learning to do robotic control tasks. Plots with "unseen-colors" in the plot title evaluate on a test set that contains colors that are unseen (i.e. out-of-distribution) with respect to the training set. Plots with "seen-colors" in the plot title evaluate on a test set that contains colors that are seen (i.e. in-distribution) with respect to the training set.

BNSL accurately extrapolates the scaling behavior of **Uncertainty Estimation / Calibration (Even Downstream)**

A.20 EXTRAPOLATION RESULTS FOR UNCERTAINTY ESTIMATION / CALIBRATION

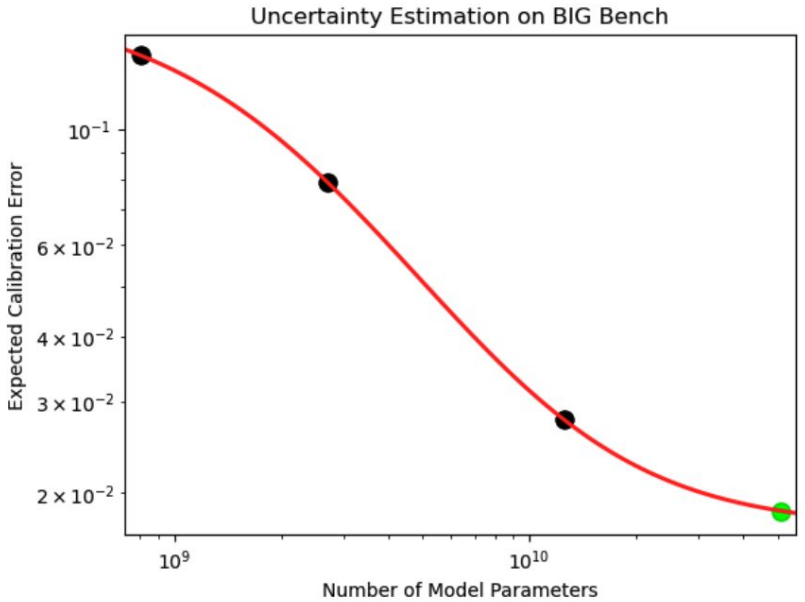


Figure 19: Extrapolation Results of BNSL for Uncertainty Estimation / Calibration. Expected Calibration Error is on the y-axis. Number of Model Parameters is on the x-axis. Experimental data obtained from "Lettered Choices (5-shot)" evaluation protocol plot from Figure 4 right of Kadavath et al. (2022). See Section A.20 for more details.

In Figure 19, we find BNSL accurately extrapolates the scaling behavior of downstream uncertainty estimation / calibration on BIG-Bench (Srivastava et al., 2022).

BNSL accurately extrapolates the scaling behavior of **Molecules**

A.27 EXTRAPOLATION RESULTS FOR MOLECULES

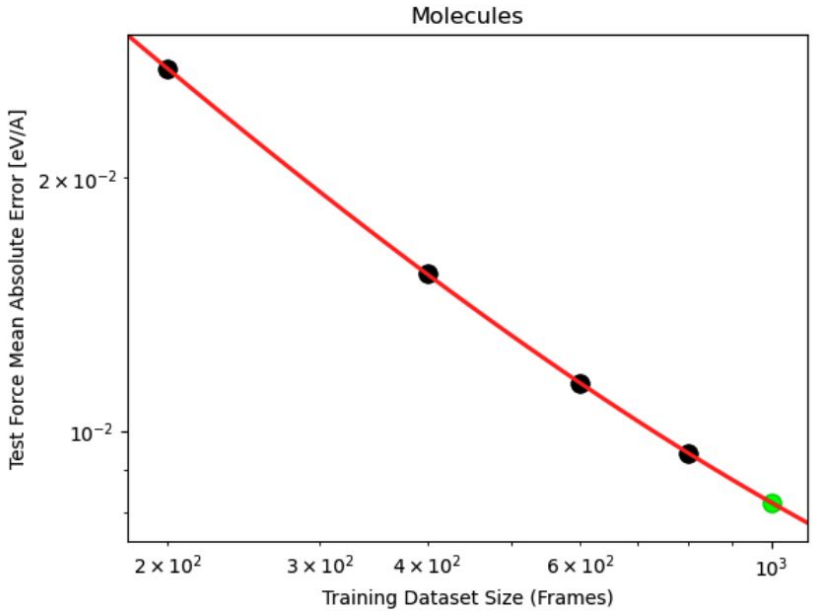


Figure 26: Extrapolation Results of BNSL for Molecules. Experimental data obtained from the "NequIP L=3" results for the aspirin molecule in MD-17 of Figure 8 of the arXiv version of Batzner et al. (2022). Y-axis is the test force mean absolute error [eV/A]. X-axis is the training dataset size (frames). See Section A.27 for more details.

In Figure 26, we find BNSL accurately extrapolates the scaling behavior of Neural Equivariant Interatomic Potentials (NequIP) graph neural networks (Batzner et al., 2022) trained via minimizing the weighted sum of energy and a force loss terms in order to predict the forces of molecules.

BNSL accurately extrapolates scaling behavior of **Audio (even Downstream)**

A.31 EXTRAPOLATION RESULTS FOR DOWNSTREAM PERFORMANCE ON AUDIO TASKS

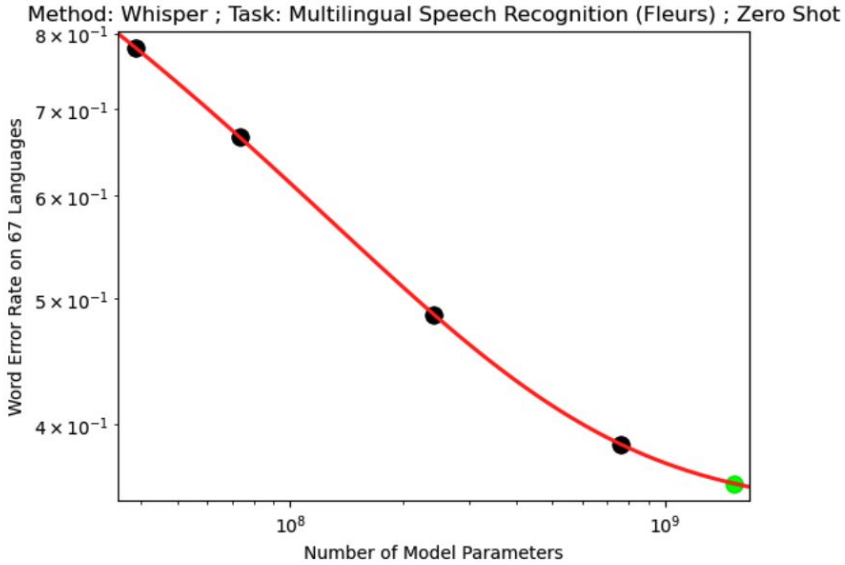


Figure 30: Extrapolation Results of BNSL for Downstream Audio Tasks when Number of Model Parameters is on the x-axis. Experimental data obtained from the second plot of Figure 6 of Whisper paper (Radford et al., 2022). The downstream task in the plot is downstream zero shot multilingual speech recognition performance on the FLEURS dataset of "Whisper" speech recognition model pretrained on a dataset of 681,070 hours of audio. See Section A.31 for more details.

In Figure 30, we show that BNSL accurately extrapolates the scaling behavior of the Downstream Performance on Audio Tasks.

BNSL accurately extrapolates the scaling behavior of **Adversarial Robustness**

A.18 EXTRAPOLATION RESULTS FOR ADVERSARIAL ROBUSTNESS

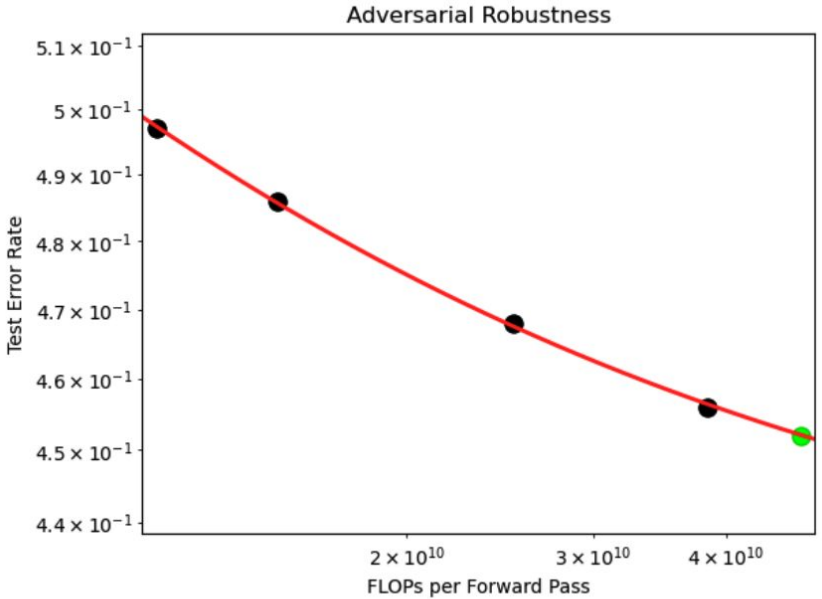


Figure 17: Extrapolation Results of BNSL for Adversarial Robustness. Test Error Rate is on the y-axis. FLOPs of the forward pass of a model of that size is on the x-axis. Experimental data of y-axis is obtained from Table 7 of Xie & Yuille (2020); experimental data of x-axis is obtained from Figure 7 of Xie & Yuille (2020). See Section A.18 for more details.

In Figure 17, we find BNSL accurately extrapolates the scaling behavior of adversarial robustness. The adversarial test set is constructed via a projected gradient descent (PGD) attacker (Madry et al., 2018) of 20 iterations. During training, adversarial examples for training are constructed by PGD attacker of 30 iterations.

BNSL accurately extrapolates the scaling behavior of **Out-of-Distribution Detection**

A.26 EXTRAPOLATION RESULTS FOR OUT-OF-DISTRIBUTION DETECTION

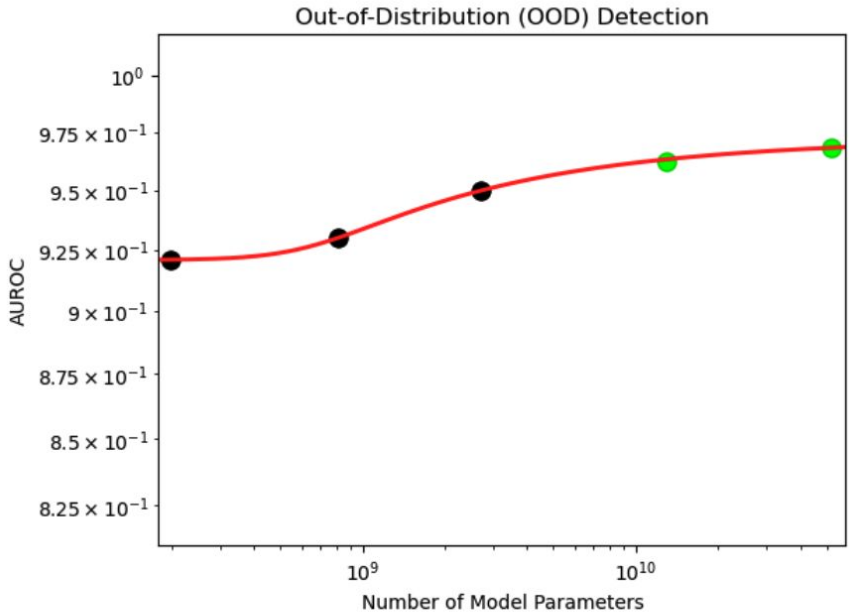


Figure 25: Extrapolation Results of BNSL for Out-of-Distribution Detection. Number of model parameters is on the x-axis. Y-axis is AUROC. Experimental data obtained from the Outlier Exposure results from Figure 23 of Bai et al. (2022) when exposed to 30 outlier examples. See Section A.26 for more details.

In Figure 25, we find BNSL accurately extrapolates the scaling behavior of Out-of-Distribution Detection performance.

BNSL accurately extrapolates the scaling behavior of **Computer Coding / Programming (even Downstream)** (even with finetuning dataset size on the x-axis)

A.19 Extrapolation Results with Finetuning Dataset Size on the X-axis (and also for Computer Programming / Coding)

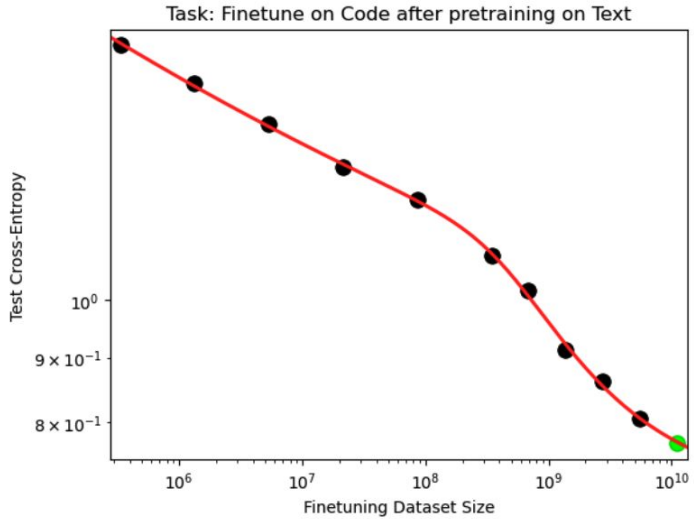


Figure 18: Extrapolation Results of BNSL with Finetuning Dataset Size on the X-axis. Experimental data is obtained from Figure 1 of Hernandez et al. (2021). The figure is of a transformer model that is pretrained on a large amount of mostly English text from the internet and then finetuned on a large amount of python code. The y-axis is Test Cross-Entropy on the distribution of python code. The x-axis is the size (measured in number of characters) of the Finetuning (not pretraining) Dataset. See Section A.17 for more details.

In Figure 18, we find BNSL accurately models and extrapolates the scaling behavior with finetuning dataset size on the x-axis (i.e. model that is pretrained on a large amount of mostly english text from the internet and then finetuned on a large amount of python code).

BNSL accurately extrapolates the scaling behavior of **Math Word Problems (even Downstream)**

A.28 EXTRAPOLATION RESULTS FOR MATH WORD PROBLEMS

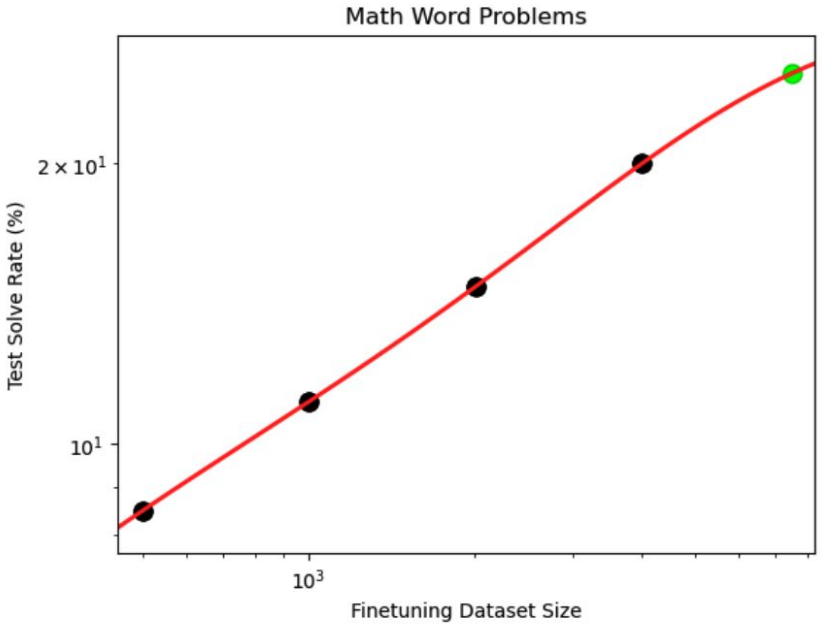


Figure 27: Extrapolation Results of BNSL for Math Word Problems. Experimental data obtained from the 12 billion parameter model results in Figure 2 left of Cobbe et al. (2021a). Y-axis is the test solve rate. X-axis is the finetuning dataset size. See Section A.28 for more details.

In Figure 27, we find BNSL accurately extrapolates the scaling behavior of large language models finetuned to solve math word problems.

BNSL accurately extrapolates the scaling behavior of **Sparse Models**

A.17 EXTRAPOLATION RESULTS FOR SPARSE MODELS

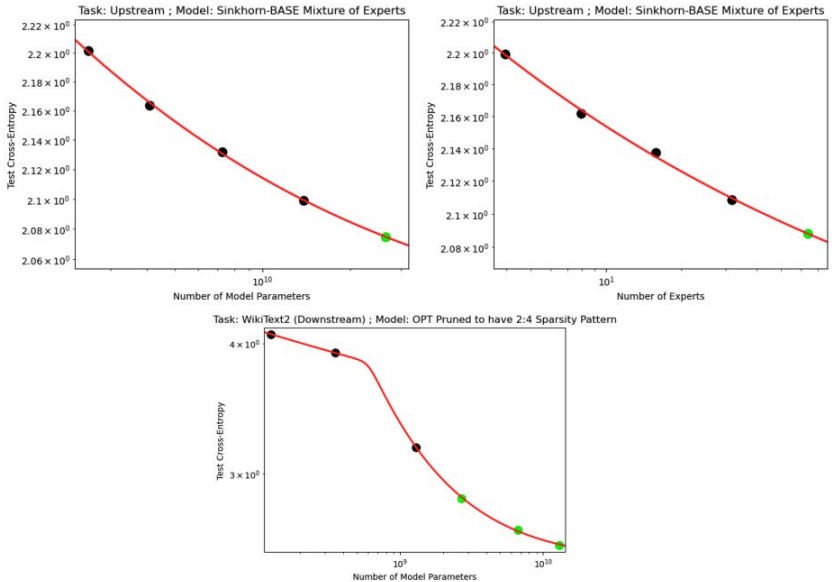


Figure 16: Extrapolation Results of BNSL for Sparse Models. Experimental data of top 2 figures are obtained from Figure 22 of Clark et al. (2022). Experimental data of bottom figure obtained from Figure 1 right of Frantar & Alistarh (2023). The y-axis is Test Cross-Entropy. The x-axis is the number of model parameters that the model contains. See Section A.17 for more details.

In Figure 16, we find BNSL accurately extrapolates the scaling behavior of various sparse models (i.e. sparse, pruned models and sparsely gated mixture-of-expert models).

BNSL accurately extrapolates the scaling behavior of **Distillation**

A.25 EXTRAPOLATION RESULTS FOR DISTILLATION

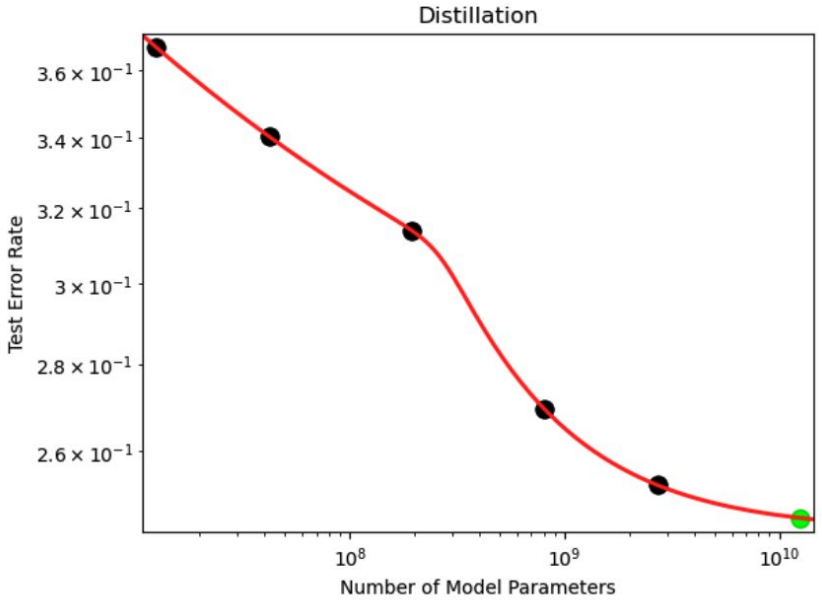


Figure 24: Extrapolation Results of BNSL for Distillation. Experimental data obtained from the Context Distillation results from Figure 5 left of Bai et al. (2022). In this setup, a language model (with the number of model parameters on the x-axis of this figure) that has been prompted is distilled into a language model (with the number of model parameters on the x-axis of this figure). Y-axis is test error rate on the helpful honest harmless (HHH) evaluation of Askell et al. (2021). See Section A.25 for more details.

In Figure 24, we find BNSL accurately extrapolates the scaling behavior of distillation.

BNSL accurately extrapolates the scaling behavior of **Quantization (even Downstream)**

A.24 EXTRAPOLATION RESULTS FOR QUANTIZATION

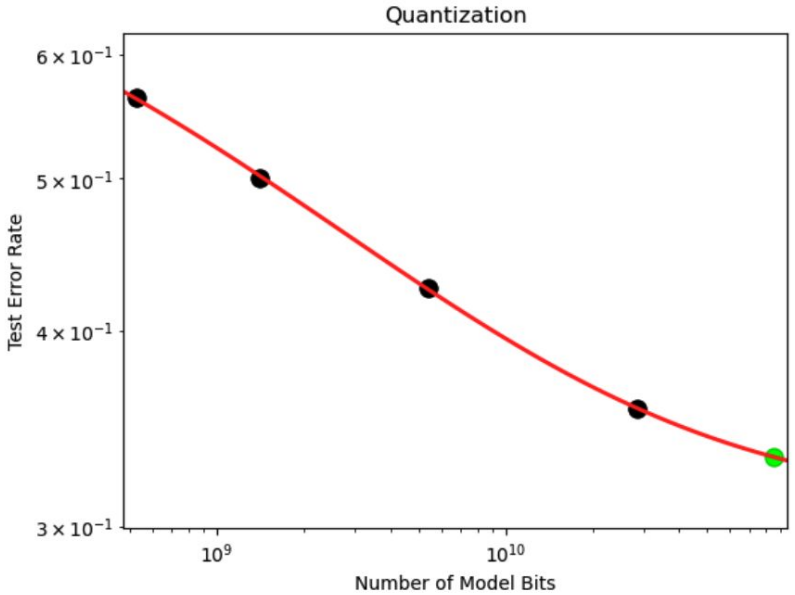


Figure 23: Extrapolation Results of BNSL for Quantization. Experimental data obtained from the 4 Bit Pythia (blockwise 64) results from Figure 8 bottom of Dettmers & Zettlemoyer (2022) in which an originally 16 bits (per parameter) model has been quantized to be 4 bits (per parameter) model. Y-axis is mean downstream zero-shot test error rate across Lambada, PiQA, Winogrande, and Hellaswag. X-axis is number of bits of parameters of model. See Section A.24 for more details.

In Figure 23, we find BNSL accurately extrapolates the scaling behavior of quantized models.

BNSL accurately extrapolates the scaling behavior of **Fairness** (and Ensembles)

A.29 Extrapolation Results for Fairness (and also Ensembles)

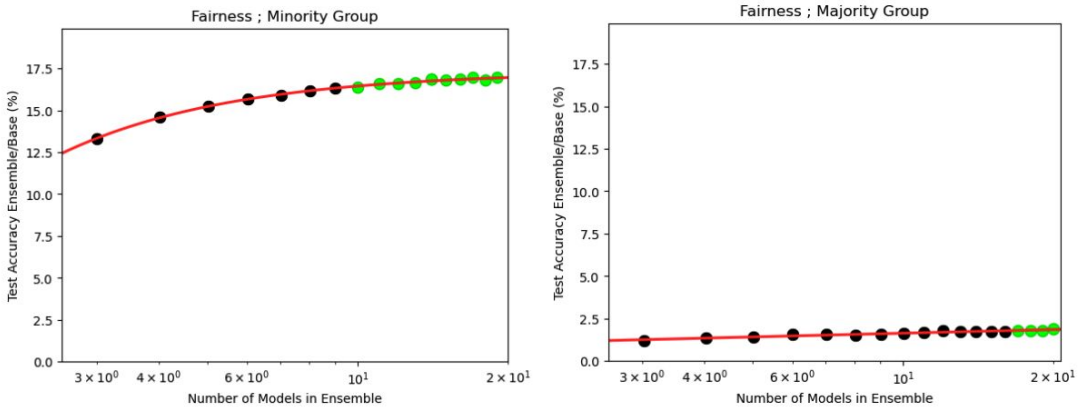


Figure 28: Extrapolation Results of BNSL for Fairness. Experimental data obtained from the Resnet-34 CIFAR-100 results in Figure 1 left of Ko et al. (2023). The model in this setup is an ensemble model. X-axis the number of models in the ensemble. Y-axis is the ratio of the ensemble's accuracy over that of a single base model. In left plot, the test dataset is the minority group which is the bottom-10 classes that are least accurately predicted. In right plot, the test dataset is the majority group which is the top-10 classes that are most accurately predicted. See Section A.29 for more details.

In Figure 28, we find BNSL accurately extrapolates the scaling behavior of fairness (and also ensembles).

BNSL accurately extrapolates the scaling behavior of **Retrieval-Augmented Models (even Downstream)**

A.15 Extrapolation Results for Retrieval-Augmented Models

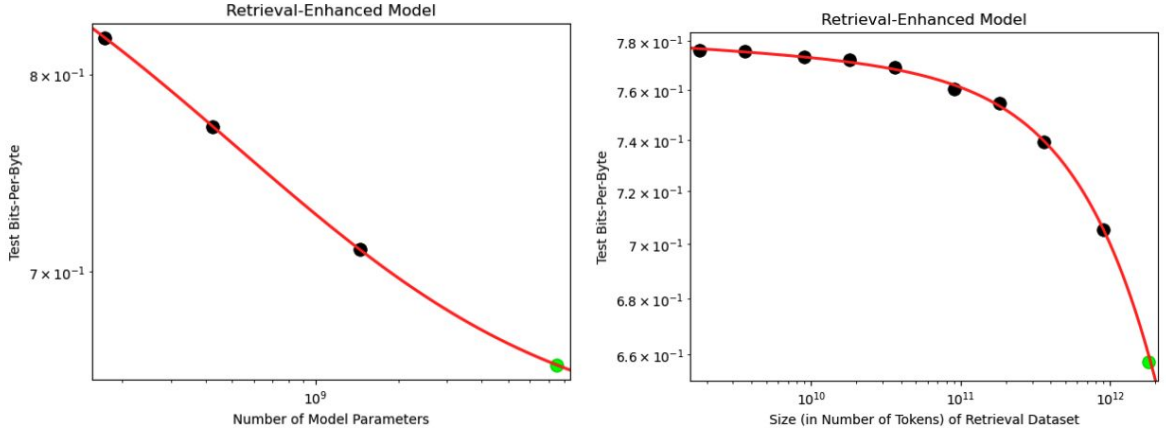


Figure 14: Extrapolation Results of BNSL for Retrieval-Augmented Models. Experimental data of left figure obtained from "RETRO [ON]" results of Figure 1 left of Borgeaud et al. (2022). Experimental data of right figure obtained from the 7.5 billion parameter model results of Figure 1 middle of Borgeaud et al. (2022). The y-axes are Zero-Shot Test Bits-per-Byte on Downstream C4 (Raffel et al., 2019) dataset. In left figure, x-axis is number of model parameters. **In right figure, x-axis notably is Size (in Number of Tokens) of the Retrieval Dataset.** See Section A.15 for more details.

In Figure 14, we find BNSL accurately extrapolates the scaling behavior of models augmented with a mechanism to retrieve data from a very large collection of data. **In right plot of Figure 14, x-axis notably is Size (in Number of Tokens) of the Retrieval Dataset.**

BNSL accurately extrapolates the scaling behavior **with Model Input Size on the x-axis (even Downstream)**

A.16 EXTRAPOLATION RESULTS WITH INPUT LENGTH (OF THE MODEL) ON THE X-AXIS

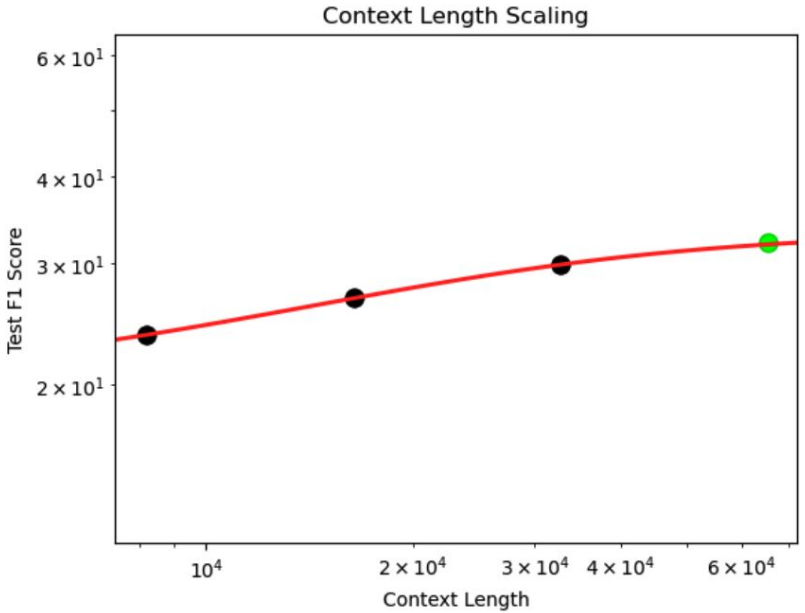


Figure 15: Extrapolation Results of BNSL for Input Size (also known as context length) of the model increases. Experimental data of obtained from the CoLT5 results of Figure 4 of Ainslie et al. (2023). The y-axis is Test F1 Score on Downstream NarrativeQA (Kočiskỳ et al., 2018) dataset. The x-axis is the is the context length (the four context length values are 8192, 16384, 32768, and 65536). See Section A.16 for more details.

In Figure 15, we find BNSL accurately extrapolates the scaling behavior with input size (also known as context length) (of the model) on the x-axis.

BNSL accurately extrapolates the scaling behavior **with Number of Training Steps on the x-axis**

A.9 EXTRAPOLATION RESULTS WITH NUMBER OF TRAINING STEPS ON THE X-AXIS

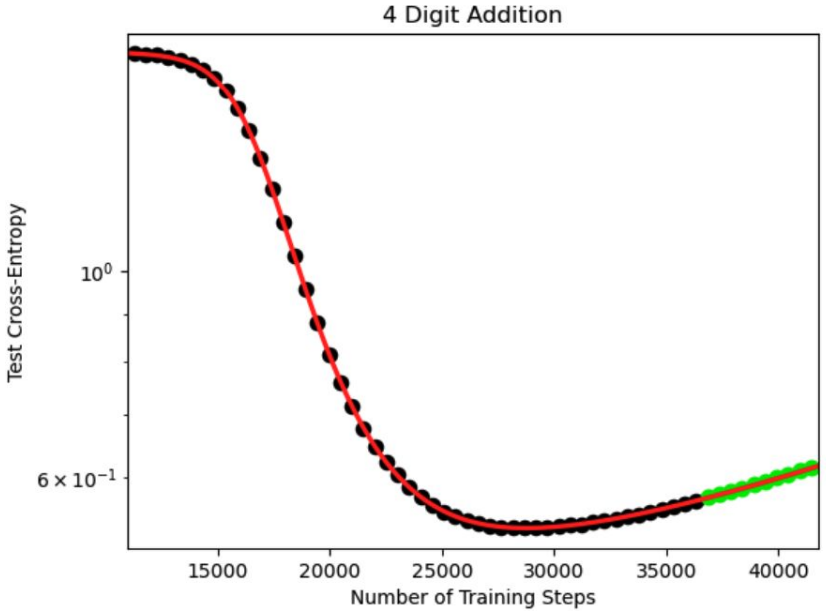


Figure 8: Extrapolation Results of BNSL with Number of Training Steps on the x-axis. The y-axis is Test Cross-Entropy. The x-axis is the number of training steps. Each point is the mean of greater than 600 seeds at that number of training steps. The training dataset size is 992. All hyperparameters and experimental details are described in Section A.5. The task is 4 digit addition. This plot contains **two breaks** of a BNSL fit to the black points. See Section A.9 for more details.

In Figure 8, we find BNSL accurately extrapolates the scaling behavior with number of training steps on the x-axis.

BNSL accurately extrapolates the downstream scaling behavior with **Upstream Performance on x-axis**

A.13 EXTRAPOLATION RESULTS WHEN UPSTREAM PERFORMANCE IS ON THE X-AXIS

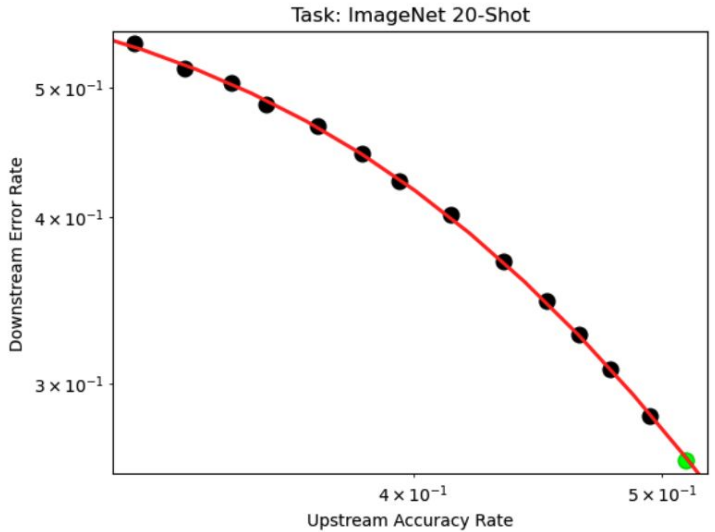


Figure 12: Extrapolation Results of BNSL for scaling behavior when Upstream Performance is on the x-axis and Downstream Performance is on the y-axis. Experimental data of scaling behavior obtained from Figure 5 of Abnar et al. (2021). The upstream task is supervised pretraining of ViT (Dosovitskiy et al., 2020) on subsets of JFT-300M (Sun et al., 2017). The Downstream Task is 20-shot ImageNet classification. See Section A.13 for more details.

In Figure 12, we show that BNSL accurately extrapolates the scaling behavior when upstream performance is on the x-axis and downstream performance is on the y-axis. The upstream task is supervised pretraining of ViT (Dosovitskiy et al., 2020) on subsets of JFT-300M (Sun et al., 2017). The downstream task is 20-shot ImageNet classification. The experimental data of this scaling behavior is obtained from Figure 5 of Abnar et al. (2021).

BNSL accurately extrapolates the scaling behavior of **Downstream Vision with amount of Compute (used for training) on the x-axis**

A.8 Extrapolation Results for Downstream Vision Tasks when training runs are scaled to be compute-optimal (and amount of compute used for (pre-)training is on the x-axis).

Task	Model	M3 ↓	BNSL ↓
ImageNet 10-Shot	ViT	1.91e-2 ± 6.48e-3	**9.79e-3 ± 4.70e-3**
ImageNet Finetune	ViT	1.14e-2 ± 2.42e-3	**9.37e-3 ± 2.60e-3**

Table 6: Extrapolation Results for Downstream Vision Tasks when training runs are scaled using the compute-optimal scaling (i.e. Pareto frontier) with respect to downstream performance. Experimental data obtained from Figure 2 of Zhai et al. (2021). See Section A.8 for more details.

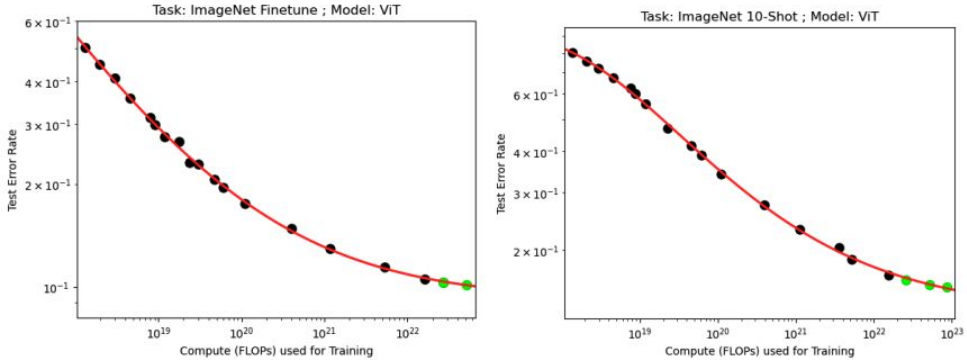


Figure 7: Extrapolation Results of BNSL for Downstream Vision Tasks when training runs are scaled to be compute-optimal. Experimental data obtained from Figure 2 of Zhai et al. (2021). See Section A.8 for more details.

In Figure 7 via fitting BNSL, we additionally obtain accurate extrapolations of scaling behavior of large-scale downstream vision tasks when compute (FLOPs) used for (pre-)training is on the x-axis and compute is scaled in the manner that is Pareto optimal with respect to the performance evaluation metric (downstream accuracy in this case). The experimental scaling data was obtained from Figure 2 of Zhai et al. (2021), and as a result in Table 6 we compare extrapolation of BNSL to the extrapolation of M3 (which was proposed in Zhai et al. (2021)); we find that BNSL that yields extrapolations of scaling behavior that are more accurate on these tasks.

BNSL also accurately extrapolates the scaling behavior **when data is pruned Pareto Optimally**

A.12 EXTRAPOLATION RESULTS WHEN DATA IS PRUNED PARETO OPTIMALLY

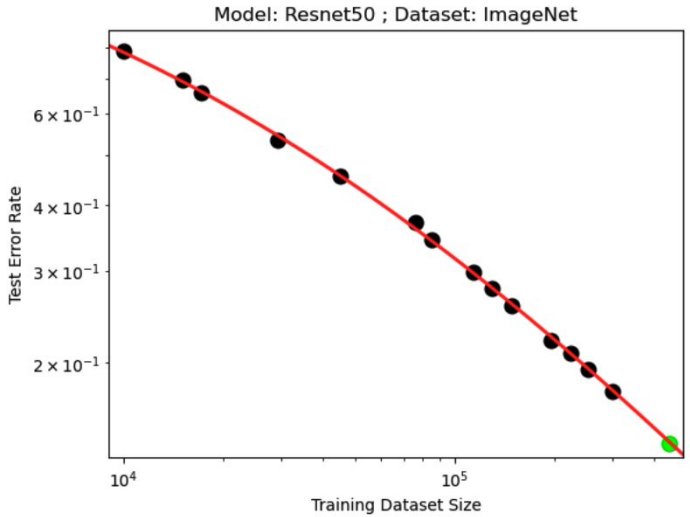


Figure 11: Extrapolation Results of BNSL for scaling behavior when data is pruned Pareto optimally (such that each point along the x-axis uses the subset of the dataset that yields the best performance (y-axis value) for that dataset size (x-axis value)). Experimental data of scaling behavior obtained from Figure 3D of Sorscher et al. (2022). See Section A.12 for more details.

In Figure 11, we show that BNSL accurately extrapolates the scaling behavior when data is pruned Pareto optimally (such that each point along the x-axis uses the subset of the dataset that yields the best performance (y-axis value) for that dataset size (x-axis value)) from Figure 3D of Sorscher et al. (2022).

BNSL accurately extrapolates the scaling behavior of **Non-Monotonic Scaling (e.g. Double Descent)**

5.4 Non-Monotonic Scaling

We show that BNSL accurately models and extrapolates non-monotonic scaling behaviors that are exhibited by Transformers (Vaswani et al. (2017)) in double descent (Nakkiran et al., 2021) in Figure 4. Various other functional forms are mathematically incapable of expressing non-monotonic behaviors (as shown in Section 4).

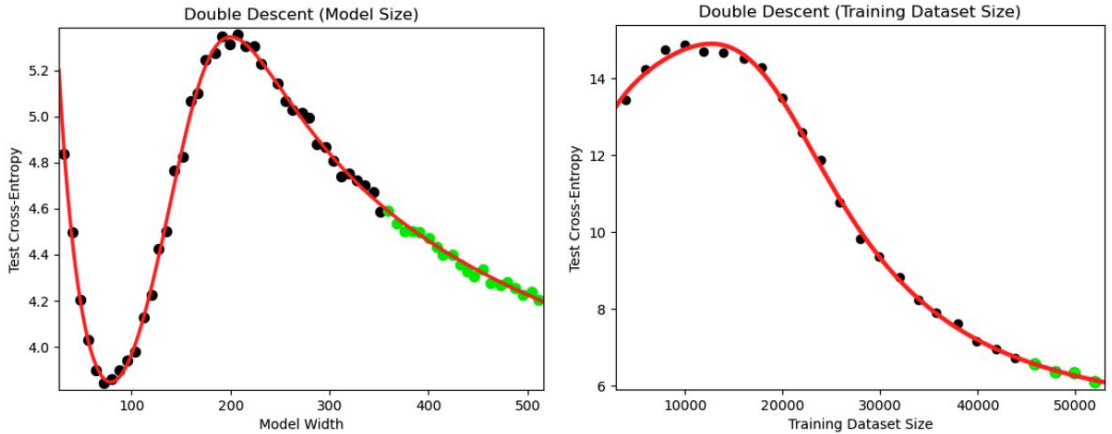


Figure 4: Extrapolation of BNSL on Double Descent. Both plots are of transformers trained to do neural machine translation via minimizing cross-entropy. Experimental data of left figure is obtained from Figure 8 top of Nakkiran et al. (2021); "Model Width" on the x-axis refers to embedding dimension d_{model} of the transformer; note that model width is linearly proportional to number of model parameters, so number of model parameters on the x-axis would yield same results. Experimental data of the right figure is obtained from Figure 11b of Nakkiran et al. (2021). The plot on the left contains **two breaks** of a BNSL fit to the black points. See Section 5.4 for more details.

BNSL accurately extrapolates the scaling behavior of **Inflection Points (a.k.a. “Emergent” “Unpredictable” “Phase Transitions”) (e.g. Four Digit Addition)**

5.5 Inflection Points

We show that BNSL is capable of modeling and extrapolating the scaling behavior of tasks that have an inflection point on a linear-linear plot such as the task of arithmetic (4-digit addition). Here we model and extrapolate the scaling behavior of a transformer model (Vaswani et al. (2017)) with respect to the training dataset size on the 4-digit addition task. Various other functional forms are mathematically incapable of expressing inflection points on a linear-linear plot (as shown in Section 4) and as a result, are mathematically incapable of expressing and modeling inflection points (on a linear-linear plot) that are present in the scaling behavior of 4-digit addition. In Figure 5 left, we show that BNSL expresses and accurately models the inflection point present in the scaling behavior of 4-digit addition and as a result accurately extrapolates the scaling behavior of 4 digit addition. For further details about the hyperparameters please refer to the Appendix Section A.5.

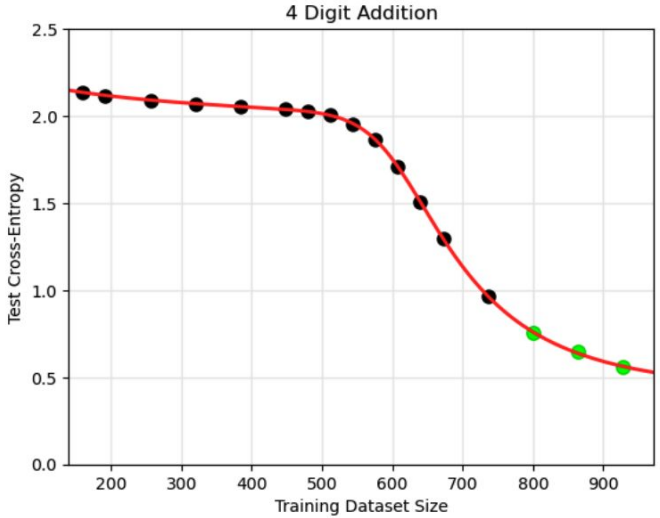


Figure 5: Extrapolation of BNSL on 4 Digit Addition. Note this plots are linear-linear. Each point is mean of greater than 1000 seeds at that dataset size. In left plot, each point is gathered from a model trained to do task of 4 digit addition. See Sections 5.5, A.5, 6, for more details.

The Limit of the Predictability of Scaling Behavior

6 The Limit of the Predictability of Scaling Behavior

We use BNSL to glean insights about the limit of the predictability of scaling behavior. Recent papers (Ganguli et al., 2022; Wei et al., 2022a) have advertised many tasks as having "unpredictable" "emergent" "phase transition/change" scaling behavior, the most famous of which is the task of arithmetic. In the previous section and in Figure 4 left, we successfully predicted (i.e. extrapolated) the scaling behavior of 4-digit addition (arithmetic). However, we are only able to accurately extrapolate the scaling behavior if given some points from training runs with a training dataset size of at least 720, and the break in which the scaling behavior of 4-digit addition transitions from one power law to another steeper power-law happens at around training dataset size of 415.

Ideally, one would like to be able to extrapolate the entire scaling behavior by fitting only points from before the break. In Figure 4 right, we use a noiseless simulation of the BNSL of 4-digit addition to show what would happen if one had infinitely many training runs / seeds to average out all the noisy deviation between runs such that one could recover (i.e. learn via a curve-fitting library such as SciPy (Virtanen et al., 2020)) the learned constants of the BNSL as well as possible. When using this noiseless simulation, we find that we are only able to accurately extrapolate the scaling behavior if given some points from training runs with a training dataset size of at least 415, which is very close to the break.

This has a few implications:

1) When the scaling behavior exhibits greater than 0 breaks that are sufficiently sharp, there is a limit as to how small the maximum (along the x-axis) of the points used for fitting can be if one wants to perfectly extrapolate the scaling behavior, even if one has infinitely many seeds / training runs.

2) If an additional break of sufficient sharpness happens at a scale that is sufficiently larger than the maximum (along the x-axis) of the points used for fitting, there does not (currently) exist a way to extrapolate the scaling behavior after that additional break.

3) If a break of sufficient sharpness happens at a scale sufficiently smaller than the maximum (along the x-axis) of the points used for fitting, points smaller (along the x-axis) than that break are often useless for improving extrapolation.

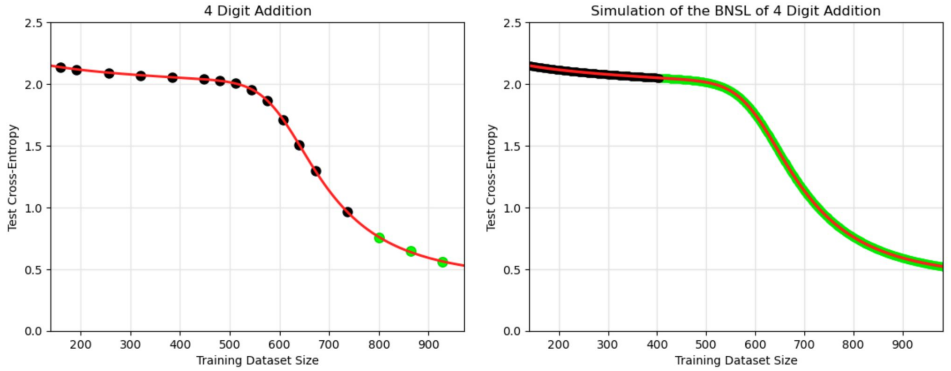


Figure 5: Extrapolation of BNSL on 4 Digit Addition. Note these plots are linear-linear. Each point in left plot is mean of greater than 1000 seeds at that dataset size. In left plot, each point is gathered from a model trained to do task of 4 digit addition. In right plot, each point is gathered from a noiseless simulation of the BNSL of the task of 4 digit addition. See Sections 5.5, A.5, 6, for more details.

Perhaps, variants of smoothly broken power laws (BNSL is an example of a variant of smoothly broken power laws) are the “true” functional form of the scaling behavior of many (all?) things that involve artificial neural networks?

Broken Neural Scaling Laws (BNSL) paper:
arxiv.org/abs/2210.14891