A Simple Model of Inference Scaling Laws

Noam Levi
Ecole Polytechnique Fédérale de Lausanne (EPFL)
Lausanne, Switzerland
`noam.levi@epfl.ch`

Abstract

Neural scaling laws have garnered significant interest due to their ability to predict model performance as a function of increasing parameters, data, and compute. In this work, we propose a simple statistical ansatz based on memorization to study scaling laws in the context of inference, specifically how performance improves with multiple inference attempts. We explore the coverage, or pass@k metric, which measures the chance of success over repeated attempts and provide a motivation for the observed functional form of the inference scaling behavior of the coverage in large language models (LLMs) on reasoning tasks. We then define an "inference loss", which exhibits a power law decay as the number of trials increases, and connect this result with prompting costs. We further test our construction by conducting experiments on a simple generative model, and find that our predictions are in agreement with the empirical coverage curves in a controlled setting. Our simple framework sets the ground for incorporating inference scaling with other known scaling laws.

1 Introduction

Advancements in deep learning have demonstrated that the performance of neural networks scales predictably as a function of model size, data size, and computational resources (Hestness et al., 2017; Kaplan et al., 2020a; Rosenfeld et al., 2020; Henighan et al., 2020a). These trends, known as *neural scaling laws*, have motivated research into understanding how scaling influences model performance in a range of domains, in particular, Large Language Models (LLMs) (Brown et al., 2020; Hoffmann et al., 2022).

However, scaling during *inference*—the process by which a trained model makes predictions on new data—has received less attention. Recent works have shown empirically that LLMs can gain substantial benefits from repeated prompts to perform better on difficult tasks such as coding and formal proofs, where verification of the correct answer can be done (Brown et al., 2024; Snell et al., 2024; Bansal et al., 2024). These works demonstrate that the performance of weaker models can be amplified without further training by repeating inference trials. A natural question then arises:

Can we interpret, or predict the inference scaling behavior of a model with repeated attempts?

To answer this question, we propose a simple toy model that isolates the *inference scaling laws* which dictate how certain performance metrics improve as a function of the number of inference attempts. Inspired by the work of Hutter (2021), which introduced a model to study scaling behavior for memorization and generalization, we devise a simple setting to capture the effect of repeated inference attempts, focusing on the *coverage* metric, also known as *pass@k*.

In this work, we present analytical predictions for coverage from a probabilistic perspective and demonstrate how inference improves with the number of repeated trials in a predictable way, which matches the observed behavior in Brown et al. (2024) and Snell et al. (2024). We use two different approaches to obtain the predicted pass@k, and highlight the connection between coverage and total inference cost. Additionally, we define a simple "inference loss", similar to the familiar test loss, but allowing for repeated trials, and predict its scaling. Lastly, we verify our predictions empirically by appealing to a simple generative model, a Variational Autoencoder (VAE), which is trained to generate reconstructions of its training data by sampling from a latent space with an

associated temperature, mimicking some of the complex generative properties of LLMs. Given that our results are isolated from the effects of other neural scaling laws, they could be incorporated into a broader exploration to find the optimal train/inference point. In particular, we hope this work sets the ground for exploring the optimal trade-off between training and inference attempts, such that the total performance, as well as cost, is minimized.

2 RELATED WORK

Scaling laws for neural networks have been extensively studied in recent years. Empirical research has shown that error rates decrease predictably as a function of increased data, parameters, and compute, following power-law relationships. Notable contributions in this space include work by Kaplan et al. (2020b), who demonstrated consistent scaling behavior across language models, and Henighan et al. (2020b), who extended these ideas to multimodal models, as well as Cabannes et al. (2024); Maloney et al. (2022); Bordelon et al. (2020); Spigler et al. (2020); Caponnetto and De Vito (2007); Steinwart et al. (2009); Fischer and Steinwart (2020); Cui et al. (2021); Levi and Oz (2023; 2024); Nam et al. (2024) who studied scaling laws for solvable yet sufficiently complex models, ranging from generalized linear regression on random feature models to kernel ridge regression, connecting them to results from random matrix theory and the underlying properties of the data. While most scaling laws focus on training, the study of *inference scaling* remains under-explored. Our work tries to fill this gap by studying how performance improves with repeated inference attempts.

3 MEMORIZING ANSATZ

In the following, we first briefly review the simplest model which produces the known data scaling law prediction by appealing to a memorizing construction, then consider our proposed model for repeated inference attempts.

The *Hutter model* (Hutter, 2021) is a probabilistic framework originally introduced to study the scaling behavior of learning curves, focusing on the relationship between memorization and generalization. It assumes a model which perfectly memorizes features during training, allowing it to correctly match sets of features and labels $\{i, y_i\}$, such that only unseen features can incur an error. The set of features is assumed to follow a Zipf power law decay with parameter α, where the probability of encountering feature i is $\theta_i \propto i^{-1-\alpha}$ and decreases with its rank. For n training samples, the expected single feature error is

$$\mathcal{E}_n = \mathbb{E}[E_i] = \sum_{i=1}^{\infty} \theta_i (1 - \theta_i)^n \approx n^{-\beta}, \qquad \beta = \frac{1}{1 + \alpha}. \tag{1}$$

Eq. (1) captures the average likelihood of encountering and labeling the feature incorrectly after n training samples. Similar to this model, we will adopt the idea of memorization as a surrogate for training, but will depart from the finite training set assumption as a basis for test errors. Furthermore, since we are only interested in successes at the single sample level, we take the index i to represent the samples themselves from here on.

3.1 PERFECT SAMPLE MEMORIZATION, IMPERFECT PREDICTIONS

In contrast to the Hutter model, we focus on a scenario where *all samples have been memorized*, hence there is no notion of test error coming from unseen data. Instead, failure during inference or data generation could arise from the need to follow a sequence of steps to reach the correct answer.

Concretely, we consider a joint model of memory M and inference I. The memory is $M \in \mathcal{M} := \mathcal{X} \rightarrow \mathcal{Y}$, where the samples are identical to the labels $\{x_i\}_{i=1}^{n} = \{y_i\}_{i=1}^{n}$, and corresponds to a model which simply learns to memorize its training data. The inference model $I \in \mathcal{H} := \mathbb{N} \rightarrow M$ takes an input index i, which serves as a proxy for a particular prompt or task, and should produce the associated memorized label (sample) y_i by recalling it from the memory M. The inference model is taken to be imperfect in its retrieval, and subject to some error ϵ, it makes predictions

$$I(i) = \begin{cases} y_i + \epsilon, & \text{with probability } p_i, \\ y_i, & \text{with probability } 1 - p_i. \end{cases} \tag{2}$$
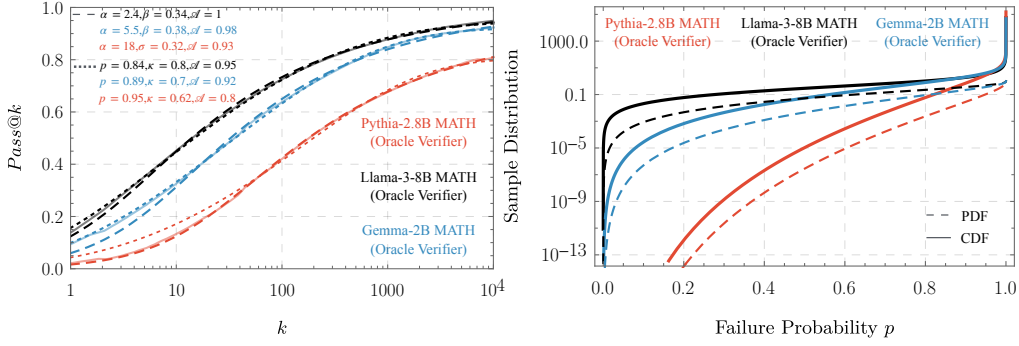
Figure 1: **Pass@k and failure distribution curves for various LLMs on difficult tasks, against theoretical scaling predictions**. *Left:* The relationship between pass@k and the number of samples for several coding and maths tasks for different models, as described in Brown et al. (2024), compared with the analytical predictions presented in Eqs. (7) and (11). The *solid* curves are data, while the *dashed* curves are the predictions from Eq. (7), where α and β correspond to the concentration of easy and hard problems, respectively. The *dotted* curves are the results of Eq. (11). The functional form in both cases captures well the LLM pass@k curves for various models, by adjusting α, β or p, κ. *Right:* The $\text{Beta}(\alpha, \beta)$ distributions for the failure probabilities are shown for the different models. We can see that most of the questions are "difficult", while the left tail behavior informs us regarding the rate of improvement with additional trials on the "easier" samples.

At this point, we only accept perfect reconstructions, such that the performance of the model on a single sample is measured simply by

$$A(i) = \mathbf{1}_{\{y_i\}}(I(i)), \tag{3}$$

where $\mathbf{1}_{\mathcal{Y}}(x) = 1$ if $x \in \mathcal{Y}$ else 0 is the indicator function.

We start from the general case where each sample has an unknown failure probability $p_i \in [0, 1]$ at inference, and we are interested in the probability of at least one successful generation of a sample over k attempts. We further assume that if a sample has been correctly generated during inference, we have access to a perfect verification method, which can determine if a single sample is correct among k possible candidates.

Assuming the attempts are independent and identically distributed (i.i.d.), the probability of failing on all k trials for a sample with failure probability p_i is simply

$$\mathbb{P}(k \text{ failures on sample } i) = \prod_{t=1}^{k}(1 - A_t(i)) = p_i^k, \tag{4}$$

where t is the trial index. Therefore, the probability of at least one success in k trials averaged over the entire dataset of size n, known as **pass@k**, is given by

$$\text{pass@k} = 1 - \frac{1}{n}\sum_{i=1}^{n}\prod_{t=1}^{k}(1 - A_t(i)) = 1 - \frac{1}{n}\sum_{i=1}^{n}p_i^k. \tag{5}$$

This will be the metric we use to describe inference accuracy for the rest of this work.

3.2 POWER LAW DISTRIBUTIONS PREDICT PASS@K FOR LLMS

To construct the distribution of failure probabilities p_i, we assume that different samples may have different inference complexity levels, incorporating some "easy" and some "difficult" samples with respect to the inference model. One way to model the different complexities is done by appealing to the so called Beta distribution. We think of $p = p_i$ as a random variable, drawn from $p \sim \text{Beta}(\alpha, \beta)$, where $\alpha \in \mathbb{R}^+$ controls how often we encounter "easier" problems, where smaller α pushes the distribution mass towards zero, while $\beta \in \mathbb{R}^+$ dictates how often we encounter "harder" problems.
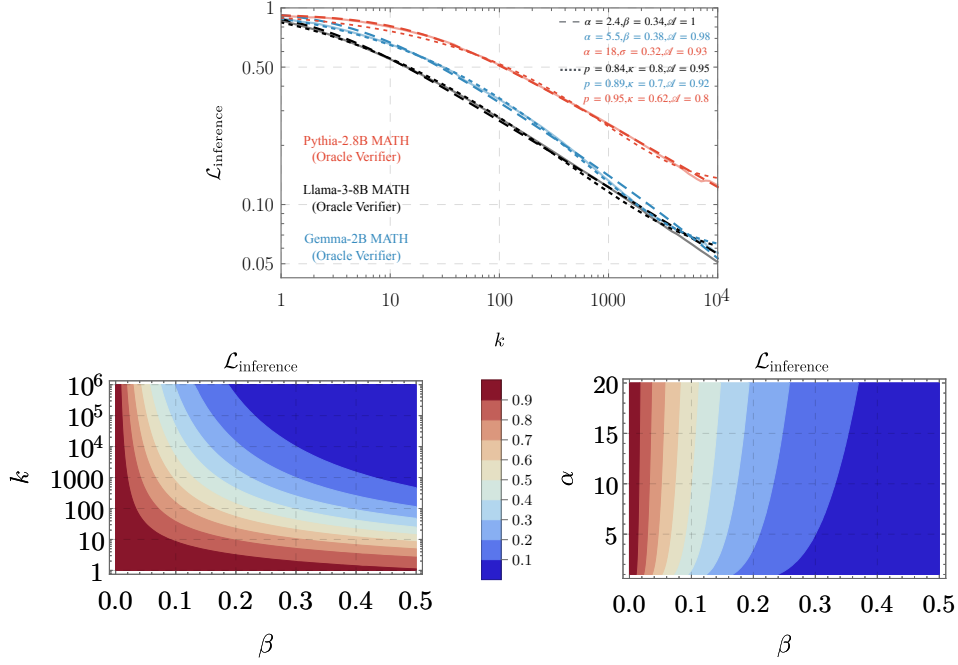
Figure 2: **Inference attempts loss $\mathcal{L}_{\text{inference}}(k)$ for repeated attempts on the memorizing model**. *Top:* Inference loss as a function of trials for the LLM experiments in Brown et al. (2024). *Bottom:* Inference loss for different β and k values. Different colors indicate inference loss values at fixed $\alpha = 5$ *(left)* and at fixed $k = 10^4$ *(right)*, illustrating the behavior of Eq. (8).

Namely, a lower β parameter increases the distribution mass towards the right tail (high failure probabilities). The Beta(α, β) PDF is given explicitly by

$$\text{Beta}(\alpha, \beta; p) = \frac{p^{-1+\alpha}(1-p)^{-1+\beta}}{B(\alpha, \beta)}, \tag{6}$$

where $B(\alpha, \beta)$ is the Euler beta function. We can therefore compute the pass@k metric by simply averaging over the failure distributions as

$$\text{pass@k} = \mathcal{A} \times \left(1 - \frac{1}{n}\sum_{i=1}^{n} p_i^k\right) \approx \mathcal{A} \times \left(1 - \langle p^k \rangle\right) = \mathcal{A} \times \left(1 - \int_0^1 dp\, p^k \frac{p^{-1+\alpha}(1-p)^{-1+\beta}}{B(\alpha, \beta)}\right)$$

$$= \mathcal{A} \times \left(1 - \frac{\Gamma(\beta)\Gamma(k+\alpha)}{B(\alpha, \beta)\Gamma(k+\alpha+\beta)}\right) \underset{k\to\infty}{\approx} \mathcal{A} \times \left(1 - \frac{\Gamma(\beta)k^{-\beta}}{B(\alpha, \beta)}\right), \tag{7}$$

where $\Gamma(z)$ is the Euler gamma function, and $\langle . \rangle$ indicates averaging over the p distribution. Since real world models such as LLMs are not perfect memorizers by our definitions, we also add an overall factor describing their maximal pass@k as $\mathcal{A}$. In Fig. 1, we show that the functional form of pass@k given in Eq. (7) is a good approximation for the empirical pass@k curves for various LLMs on mathematical and coding tasks, as reported in Brown et al. (2024).

We see that at large k, we find a power law decay of the effective average failure probability, a common feature in neural scaling laws. If we then define the **inference loss $\mathcal{L}_{\text{inference}}(k)$** as the expectation with respect to the sample distribution over errors, our results for pass@k correspond to

$$\mathcal{L}_{\text{inference}}(k) \equiv \mathbb{E}(\text{Error in } k \text{ trials}) = \mathbb{E}(\mathcal{A} \times p^k) \tag{8}$$

$$\approx \mathcal{A} \times \frac{\Gamma(\beta)\Gamma(k+\alpha)}{B(\alpha, \beta)\Gamma(k+\alpha+\beta)} \underset{k\to\infty}{\approx} \mathcal{A} \times \frac{\Gamma(\beta)k^{-\beta}}{B(\alpha, \beta)}.$$

This result implies that the model test loss with repeated inference steps will decrease mainly depending on the value of the exponent β. Intuitively, it means that for a fixed α parameter, the harder the questions appear to the model, the more inference attempts are required to reach a low loss. We illustrate this behavior in Fig. 2.

3.2.1 INTERPRETABILITY

The result in Eq. (8) not only allows one to predict a threshold for inference in order to get an increase in model performance on difficult problems, but could also offer some interpretation for the difficulty of tasks with respect to the trained model. We can gain some insights regarding the real-world models by fitting the pass@k metric according to Eq. (7) to the ones given in Brown et al. (2024), and attempt to interpret the properties of the test data from the parameters α and β. In particular, note that the ratio $\frac{\alpha}{\alpha+\beta}$ gives the mean of the Beta distribution, which represents the average failure probability across the samples. If $\alpha > \beta$, the mean failure probability is high (i.e., most samples are harder). On the other hand, if $\alpha < \beta$, the mean failure probability is low, implying that most samples are easy. Furthermore, the denominator $\alpha + \beta$ governs the concentration of the distribution, such that when $\alpha + \beta$ is large, the failure probabilities p_i are more tightly clustered around the mean (more homogeneity in difficulty), and conversely for small $\alpha + \beta$.

From Fig. 1, we can see that the typical values for the "harder" problem parameter $\beta \sim 0.35$, while the tail parameter is $2 < \alpha < 20$. This implies that most of the problems in the datasets used to measure the pass@k curves were indeed difficult in terms of the model's perception, while the existence of a left tail implies that the easy samples are covered quickly with less than 100 trials. This is reflected in the right panel of Fig. 1.

Another point of interest is that the average performed in Eq. (7) can be thought of as a Laplace transform from the variable $\sigma = \log(1/p)$ to the trials space k, in the sense that

$$\tilde{f}(k) = \langle p^k \rangle = \int_0^\infty d\sigma e^{-\sigma k} \frac{e^{-\alpha\sigma}(1 - e^{-\sigma})^{-1+\beta}}{B(\alpha, \beta)} = \int_0^\infty d\sigma e^{-\sigma k} f(\sigma). \tag{9}$$

This interpretation implies that it is possible to derive the probability distribution function of the samples in terms of their perceived difficulty by performing the inverse transform. In particular, given an empirical pass@k metric obtained for a given model, the inverse transform on $\tilde{f}(k) = (\mathcal{A} - \text{pass@}k)/\mathcal{A}$ will yield the perceived difficulty PDF. Potentially, such a procedure can be used to identify "difficult" and "easy" questions and construct improved fine-tuning algorithms by choosing training samples biased towards the "difficult" but obtainable tasks.

3.3 CORRELATED TRIALS AND EFFECTIVE k APPROACH

In the previous section, we showed that correlated samples drawn iid from a varying complexity distribution can effectively describe the pass@k metric for memorizing models. Here, we take a converse approach, where do not assume that samples are correlated through their failure rate distribution, but instead that *trials* themselves are correlated.

One can conjecture that dependencies between trials arise due to the internal model structure and the data itself. To capture these correlations, we suggest a model where the correlation between trials decays as a power law, implying that successive trials become less independent as we increase the number of trials.

In order to incorporate the correlation between trials, we define the notion of an *effective number of independent trials*, denoted k_{eff}. This adjusts the original k to account for the decay in trial independence. The correlation between trials is modeled via a power-law decay in the eigenvalues of the correlation matrix, such that the effective number of independent trials is given by

$$k_{\text{eff}} = \sum_{i=1}^k i^{-\kappa} = H_k(\kappa) \underset{k \to \infty}{\approx} \left(\frac{1}{2} - \frac{k}{\kappa - 1}\right)k^{-\kappa} + \zeta(\kappa), \qquad \kappa \in \mathbb{R}_{\geq 0} \tag{10}$$

where κ is the power-law exponent governing how quickly the correlation between trials decays, $H_k(\kappa)$ is the Harmonic number, and $\zeta(\kappa)$ is the Riemann zeta function.

Thus, the probability of at least one success in k trials, incorporating correlations, becomes

$$\text{pass@}k = \mathcal{A} \times \left(1 - \frac{1}{n}\sum_{i=1}^n p_i^{k_{\text{eff}}}\right) = \mathcal{A} \times \left(1 - p^{H_k(\kappa)}\right) \approx \mathcal{A} \times \left(1 - p^{\left(\frac{1}{2} - \frac{k}{\kappa-1}\right)k^{-\kappa} + \zeta(\kappa)}\right), \tag{11}$$

where $p = p_i$ is the error probability of every sample, and k_{eff} accounts for correlations between trials. The result of Eq. (11) is shown in Fig. 1 as the dotted curves, which approximate the LLM
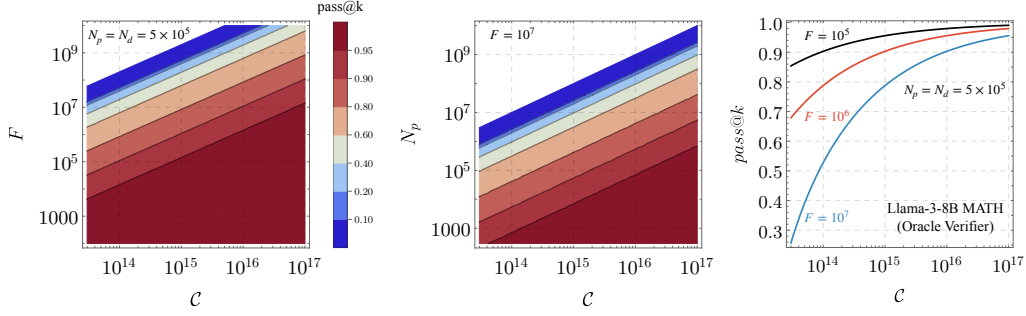
Figure 3: **Pass@k as a function of total inference cost for Llama-3-8B MATH (Oracle Verifier)**. *Left and Center:* We show the pass@k metric as a function of number of total inference cost and number of FLOPS per token F or number of prompt/decode tokens $N_p = N_d$ in $\log, \log$. We see that there is a clear trade-off between total inference cost whenever keeping one of the parameters fixed, in a predictable way from Eq. (13). *Right:* We show a slice of the contour plots for fixed $N_p = N_d$, and changing the number of FLOPS per token. The parameters chosen for these figures are fitted from Eq. (7) applied to the data taken from Brown et al. (2024).

behavior well for $k \gg 1$. We stress that this approach should be taken as an effective description, which nevertheless manages to accurately capture the same behavior as sample correlations.

3.4 CONNECTION TO COMPUTE SCALING

Here, we would like to translate our results from the attempts variable to the inference cost. A natural proxy for the inference cost may be the number of required Floating Point Operations Per Second (FLOPS)[1]. For concreteness, we adapt the total inference cost formula suggested in Brown et al. (2024), given by

$$\text{total inference FLOPS} \approx \text{num prompt tokens} \times \text{FLOPS per token} \tag{12}$$
$$+ \text{ num decoded tokens} \times \text{FLOPS per token} \times \text{num completions}$$
$$\rightarrow \mathcal{C} = N_p \times F + N_d \times F \times k,$$

where $\mathcal{C}$ is the total inference cost, N_p, N_d are the number of prompt and decode tokens, respectively, and F is the number of FLOPS per token. We can convert some of our pass@k results to this metric by taking the large k limit of Eq. (7), giving

$$k = \left(\frac{(\mathcal{A} - \text{pass@k})B(\alpha, \beta)}{\mathcal{A}\Gamma(\beta)} \right)^{-1/\beta} \rightarrow \text{Coverage}(\mathcal{C}) \approx \mathcal{A} \times \left(1 - \frac{\Gamma(\beta)}{B(\alpha, \beta)} \left(\frac{\bar{\mathcal{C}} - N_p}{N_d} \right)^{-\beta} \right), \tag{13}$$

where $\bar{\mathcal{C}} \equiv \mathcal{C}/F$ is the normalized total inference cost. Keeping all uncontrollable parameters fixed, such as the number of FLOPS per token fixed, we see from Fig. 3 that it is worthwhile to reduce the number of prompting and decoding tokens or increase the number of attempts, up to the minimal amount of total inference compute required to reach a target coverage. The numbers in Fig. 3 are chosen to mimic the results found in Brown et al. (2024), and are only meant to give an illustration of the functional behavior of Eq. (13).

Alternatively, we can phrase Eq. (13) in terms of the inference loss as

$$\mathcal{L}_{\text{inference}}(\mathcal{C}) \approx \mathcal{A} \times \frac{\Gamma(\beta)}{B(\alpha, \beta)} \left(\frac{\bar{\mathcal{C}} - N_p}{N_d} \right)^{-\beta}, \tag{14}$$

which demonstrates the power law decay of the inference loss with total inference cost, depending on the value of β.

---

[1]This may be too crude a cost metric that ignores other aspects of system efficiency (Dehghani et al., 2021), which may benefit from repeated sampling (Juravsky et al., 2024; Zheng et al., 2023), as pointed out in Brown et al. (2024).
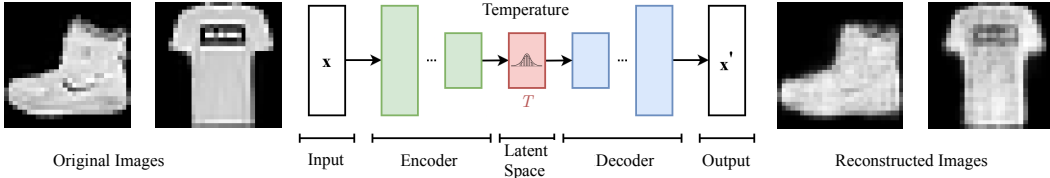
Figure 4: **Visualization of the task described in Section 4**. Here, a VAE is tasked with generating samples from its training data, where a "failure" occurs when the reconstruction error falls above a certain threshold ϵ.
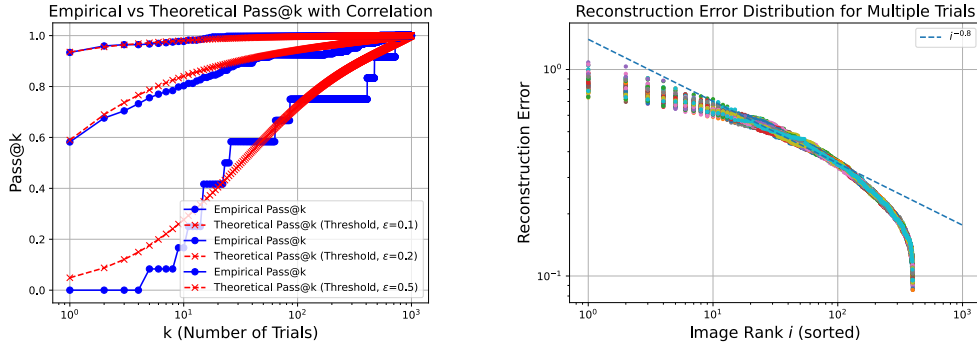

Figure 5: **Results for the VAE reconstruction task, compared with semi-analytical predictions**. *Left:* The pass@k metric as a function of number of attempts k, for different threshold values, with temperature $T = 1.1$. The curves have been normalized to asymptote at 1 for visual clarity. *Right:* The reconstruction error behavior across multiple trials, indicated by different colors. The errors obey a quasi power law behavior.

4   EXPERIMENTS ON A SIMPLE GENERATIVE MODEL

To further validate some of our analytical understanding in a controllable setting, we perform a series of experiments in which we train a simple generative model to reconstruct images taken from Fashion-MNIST (Xiao et al., 2017). Our goal is to connect the theoretical memorizing model and the behavior of more complex generative models by attempting to accurately reconstruct "memorized" examples, heuristically shown in Fig. 4.

To do this, we train a variational autoencoder (VAE) with a temperature parameter to study how errors propagate over multiple trials and to compare empirical pass@k with theoretical predictions under correlated trials. We refer to this as the *VAE reconstruction task*.

To quantify the error probability of the model over multiple samples, we define the error per sample using the norm of the difference between the reconstructed and original image:

$$\text{error}(i) = \frac{\|\hat{y}_i - y_i\|}{\|y_i\|}. \tag{15}$$

Here, y_i represents the original image, and $\hat{y}_i$ is the reconstruction. This per-sample error metric allows us to define success or failure at the sample level, where a trial is considered successful if the reconstruction error falls below a threshold ϵ.

To empirically calculate pass@k, we sample multiple reconstructions from the VAE for each input sample. For each sample, we conduct k trials and calculate whether at least one trial resulted in a reconstruction with error less than some chosen threshold value $\epsilon \in [0, 1]$. Pass@k is then computed as the fraction of samples for which the model succeeded to reconstruct at least once in k trials.

In Fig. 5, we show the pass@k results for the VAE reconstruction task, for different threshold ϵ values (left), as well as the reconstruction error distribution for multiple attempts (right). The theoretical predictions for the pass@k curves are shown for the effective k approach, given in Eq. (11),
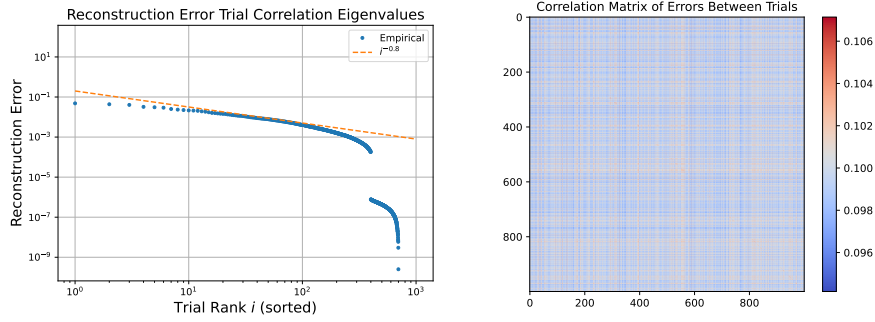
Figure 6: **Correlation matrix for errors across different trials for the VAE reconstruction task.**
Left: The eigenvalues of the correlation matrix follow a power law decay with the number of trials.
Right: Visualization of the correlation matrix itself shows clear correlations between different trials
for $k = 1000$.

and approximate the VAE results well. Here, p_i is computed empirically by taking the distribution
at the maximal k and κ is taken from the correlation matrix in Fig. 6.

To complete the picture, we empirically confirm that the assumption of independent trials is indeed
violated, as trials are effectively correlated. To capture this effect, we compute the correlation matrix
for the errors across trials $\epsilon_{kk'}$ as

$$\epsilon_{kk'} = \frac{1}{n} \sum_i \text{error}_{i,k} \times \text{error}_{i,k'}. \tag{16}$$

The eigenvalues of the correlation matrix decay as a power law, suggesting that the effective number
of independent trials diminishes as k increases, which is clearly depicted in Fig. 6 (left).

5 CONCLUSIONS

In this paper, we have proposed a simple statistical explanation for a so-called inference scaling law,
which describes the scaling of the coverage (pass@k) metric with number of repeated attempts for
generative models. We presented two possible models which lead to inference scaling: One based
on introducing a sample space distribution of "easy" and "difficult" problems, and the other on an
effective Zipf-like correlation structure between trials. Using these simple constructions, we were
able to derive analytical predictions for the pass@k, as well the test loss as a function of repeated
attempts, which we dubbed the inference loss. We then verified our predictions empirically both
through previous experimental results for LLMs and for a simple generative VAE construction.

We stress that the merit of our construction is in its simplicity, and there are many other models who
can give rise to the same functional behavior. We view this as a positive rather than a negative, since
it means that this simple model captures a universal behavior, which should not depend much on
the modeling itself. For instance, another way to arrive at a similar scaling law would be to choose
a different modeling for the failure distribution, based perhaps on program length, and introducing
the notion of a distribution of program lengths corresponding to different samples, similar to Ringel
and de Bem (2018). In the end, this type of construction will have a similar interpretation in terms
of task complexity w.r.t the model.

We believe our toy model offers a simple yet effective phenomenological framework for under-
standing how inference quality improves with more opportunities to predict correctly. Future work
could extend this framework to more complex models, including applying similar methodology as
Maloney et al. (2022) to generalized linear regression, kernel regression and neural networks, and
investigate how it interacts with existing scaling laws based on model size and training data.

REFERENCES

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020a.

Jonathan S Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. In *International Conference on Learning Representations*, 2020.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020a.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, pages 30016–30030, 2022.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling, 2024. URL https://arxiv.org/abs/2407.21787.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL https://arxiv.org/abs/2408.03314.

Hritik Bansal, Arian Hosseini, Rishabh Agarwal, Vinh Q. Tran, and Mehran Kazemi. Smaller, weaker, yet better: Training llm reasoners via compute-optimal sampling. 2024. URL https://api.semanticscholar.org/CorpusID:272146630.

Marcus Hutter. Learning curve theory, 2021. URL https://arxiv.org/abs/2102.04074.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020b. URL https://arxiv.org/abs/2001.08361.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. Scaling laws for autoregressive generative modeling, 2020b. URL https://arxiv.org/abs/2010.14701.

Vivien Cabannes, Elvis Dohmatob, and Alberto Bietti. Scaling laws for associative memories, 2024. URL https://arxiv.org/abs/2310.02984.

Alexander Maloney, Daniel A Roberts, and James Sully. A solvable model of neural scaling laws. *arXiv preprint arXiv:2210.16859*, 2022.

Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. In *International Conference on Machine Learning*, pages 1024–1034. PMLR, 2020.

Stefano Spigler, Mario Geiger, and Matthieu Wyart. Asymptotic learning curves of kernel methods: empirical data versus teacher–student paradigm. *Journal of Statistical Mechanics: Theory and Experiment*, (12):124001, 2020.

Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7:331–368, 2007.

Ingo Steinwart, Don R Hush, Clint Scovel, et al. Optimal rates for regularized least squares regression. In *COLT*, pages 79–93, 2009.

Simon Fischer and Ingo Steinwart. Sobolev norm learning rates for regularized least-squares algorithms. *The Journal of Machine Learning Research*, 21(1):8464–8501, 2020.

Hugo Cui, Bruno Loureiro, Florent Krzakala, and Lenka Zdeborová. Generalization error rates in kernel regression: The crossover from the noiseless to noisy regime. In *Advances in Neural Information Processing Systems*, volume 34, pages 10131–10143, 2021.

Noam Levi and Yaron Oz. The universal statistical structure and scaling laws of chaos and turbulence, 2023. URL https://arxiv.org/abs/2311.01358.

Noam Levi and Yaron Oz. The underlying scaling laws and universal statistical structure of complex datasets, 2024. URL https://arxiv.org/abs/2306.14975.

Yoonsoo Nam, Nayara Fonseca, Seok Hyeong Lee, Chris Mingard, and Ard A. Louis. An exactly solvable model for emergence and scaling laws in the multitask sparse parity problem, 2024. URL https://arxiv.org/abs/2404.17563.

Mostafa Dehghani, Anurag Arnab, Lucas Beyer, Ashish Vaswani, and Yi Tay. The efficiency misnomer. *CoRR*, abs/2110.12894, 2021. URL https://arxiv.org/abs/2110.12894.

Jordan Juravsky, Bradley Brown, Ryan Ehrlich, Daniel Y. Fu, Christopher R'e, and Azalia Mirhoseini. Hydragen: High-throughput llm inference with shared prefixes. *ArXiv*, abs/2402.05099, 2024. URL https://api.semanticscholar.org/CorpusID:267523163.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Jeff Huang, Chuyue Sun, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. Efficiently programming large language models using sglang, 2023.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. URL https://arxiv.org/abs/1708.07747.

Zohar Ringel and Rodrigo de Bem. Critical percolation as a framework to analyze the training of deep networks, 2018. URL https://arxiv.org/abs/1802.02154.
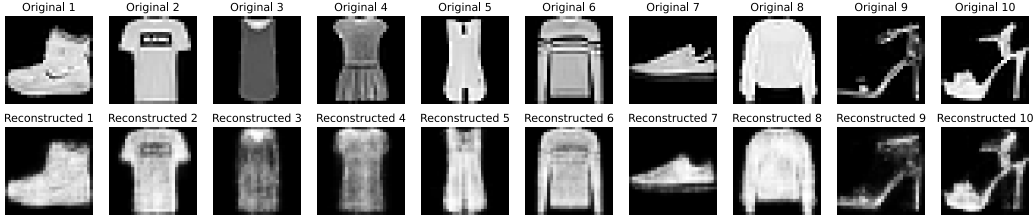
A EXPERIMENTAL SETUP

Figure 7: **Examples of original and reconstructed samples from the VAE reconstruction task.**

Here, we specify the experimental procedure used for the analysis in Section 4.

We utilize a Variational Autoencoder (VAE) with the following architectural details:

- **Input dimension:** $28 \times 28 = 784$, corresponding to the flattened pixel values of the Fashion MNIST dataset.
- **Hidden dimension:** 400.
- **Latent dimension:** 20, controlling the bottleneck for information in the latent space.
- **Decoder:** The decoder reconstructs the original input through two fully connected layers, outputting a 784-dimensional vector followed by a sigmoid activation to ensure pixel values remain between 0 and 1.
- **Temperature parameter:** A temperature parameter $T = 1.1$ is applied during the reparameterization step to control the variance of the latent variables, allowing us to model uncertainty in the latent space more effectively.

The VAE was trained on the first 400 samples from the Fashion MNIST dataset. The loss function combines binary cross-entropy for reconstruction and the Kullback-Leibler divergence to regularize the latent variables. We ran the training for 1000 epochs using the Adam optimizer with a learning rate of 1×10^{-3}.

To provide qualitative insights into the model's performance, we visualize several input samples from the Fashion-MNIST dataset along with their corresponding reconstructions in Fig. 7. This allows us to inspect both successful and failed reconstructions, and examine the types of errors the model makes.