Densing Law of LLMs

Chaojun Xiao[1], **Jie Cai**[2], **Weilin Zhao**[1], **Guoyang Zeng**[2], **Biyuan Lin**[2], **Jie Zhou**[2], **Zhi Zheng**[2]
Xu Han[1], **Zhiyuan Liu**[1], **Maosong Sun**[1]
[1]Tsinghua University [2]ModelBest Inc.
xiaocj20@mails.tsinghua.edu.cn
{han-xu,liuzy,sms}@tsinghua.edu.cn

Highlights

We introduce the concept of "capability density" to evaluate the training quality of large language models (LLMs) and describe the trend of LLMs that considers both effectiveness and efficiency.

(Relative) Capability Density. *For a given LLM $\mathcal{M}$, its capability density is defined as the ratio of its **effective parameter size** to its actual parameter size, where the effective parameter size is the minimum number of parameters required for the reference model to achieve performance equivalent to $\mathcal{M}$.*

We reveal an empirical law for the capability density of ***open-source base LLMs*** released since 2023.

Densing Law. *The maximum capability density of LLMs exhibits an exponential growth trend over time.*

$$ln(\rho_{max}) = At + B$$

Here, ρ_{max} is the maximum capability density of LLMs at time t.

Figure 1 presents the capability density of popular LLMs, measured by their performance on 5 widely-used benchmarks. A trend is fitted between maximum capability density and release date, revealing that $A \approx 0.007$ with $R^2 \approx 0.93$. This indicates **the maximum capability density of LLMs doubles approximately every 3.3 months**[1]. That means, around three months, it is possible to achieve performance comparable to current state-of-the-art LLMs using a model with half the parameter size.

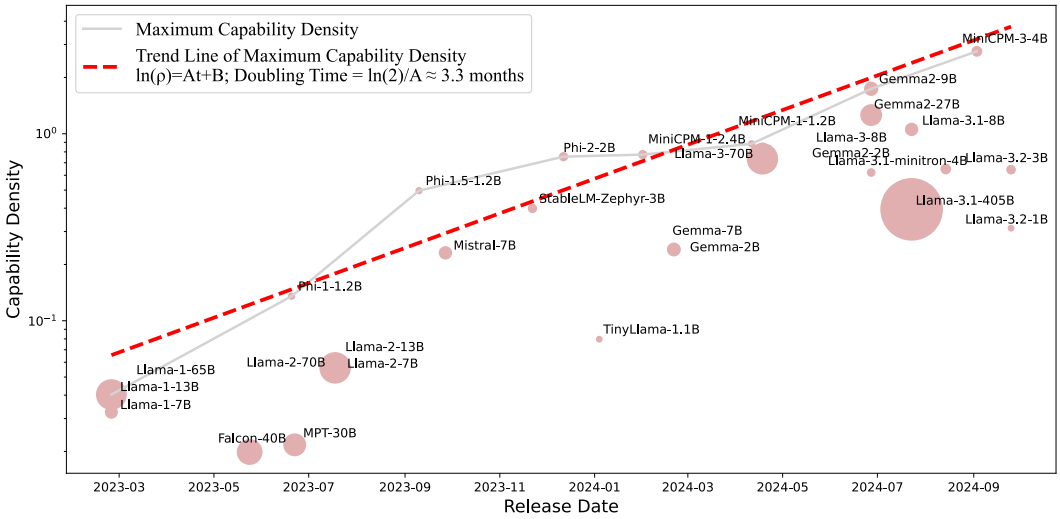


Figure 1: The estimated capability density of open-source base LLMs.

[1]The capability density growth rate is affected by specific evaluation benchmarks and reference models.

the scaling function, for any given model, we calculate its effective parameter size – the number of parameters the reference model would need to achieve equivalent performance. The density of an LLM relative to the reference model is then defined as the ratio of its effective parameter size to its actual parameter size. By introducing the concept of model density, we aim to more accurately measure model quality and enable comparisons between models of different scales. This evaluation method has the potential to provide new insights into the future direction of LLM development, helping researchers find the optimal balance between effectiveness and efficiency.

1.1 Key Findings

After defining LLM density, we analyze 29 widely-used *open-source pre-trained base models* from recent years. Our key finding for model density is:

Densing Law. *The maximum capability density of LLMs exhibits an exponential growth trend over time.*

$$ln(\rho_{max}) = A \cdot t + B$$

Here, ρ_{max} is the maximum capability density of LLMs at time t.

Based on our evaluation on 5 widely-used benchmarks, MMLU (Hendrycks et al., 2020), BBH (Suzgun et al., 2023), MATH (Hendrycks et al., 2021), HumanEval (Chen et al., 2021), and MBPP (Austin et al., 2021), $A \approx 0.007$, which means the maximum density of LLMs doubles approximately every three months. For example, MiniCPM-1-2.4B released on February 1st, 2024, can achieve comparable or even superior performance with Mistral-7B released on September 27th, 2023. We can use an LLM with only 35% parameters to obtain roughly equivalent performance after 4 months. It is worth noting that using different evaluation benchmarks may result in slight variations in the estimation and growth rate of model density. We encourage the community to develop more comprehensive evaluation benchmarks for LLMs to ensure more accurate measurements of density.

Based on the conclusion that the density of LLMs is continuously increasing in an exponential trend, we can further deduce the following implications:

Corollary 1. ***Inference Costs Decrease Exponentially****: The inference costs are going down exponentially for LLMs with equivalent downstream performance.*

Densing Law indicates that the ratio of effective parameter size to the real parameter size doubles approximately every three months. Intuitively speaking, in three months, we can achieve performance comparable to the current state-of-the-art model using a model with only half the number of parameters. Thus, the inference costs are going down exponentially for equivalent downstream performance. We find that from January 2023 to the present, the inference cost of GPT-3.5-level models has decreased by 266.7 times.

Corollary 2. ***Densing Law*** $\times$ ***Moore's Law****: The effective parameter size of LLMs that can run on chips of the same area increases exponentially.*

Moore's Law (Moore, 1965) states that the number of circuits integrated on a chip of the same area increases exponentially. This implies an exponential increase in computing power. Densing Law indicates that the density of LLMs doubles every 3.3 months. Combining these two factors, we can conclude that the effective parameter size of LLMs that can be run on a chip of the same price increases faster than both LLMs' density and computation power of chips.

Corollary 3. ***Density Growth Accelerated after ChatGPT's Release****: With the release of ChatGPT, the growth rate of LLM density increased by 50%.*

We compare the increasing trends in LLMs' density before and after the release of ChatGPT. The results show that following the release of the ChatGPT model, the growth rate of maximum density has noticeably accelerated. Specifically, after the release of ChatGPT, the growth rate of LLM density increased by 50%.

Corollary 4. ***Efficient Compression ≠ Density Improvement****: Existing pruning and distillation methods usually cannot lead to efficient LLMs with higher density.*

To enhance model inference efficiency, many researchers have devoted efforts to a series of model compression algorithms, such as pruning and distillation (Ma et al., 2023; Sun et al., 2024; Yang et al., 2024; Xu et al., 2024). These algorithms are often believed to improve the performance of the resulting compressed models. However, by comparing some models with their compressed counterparts, we can observe that the widely used pruning and distillation methods usually result in smaller models with lower density than the original models. We encourage the community to further explore more effective model compression algorithms, with a greater emphasis on improving the density of smaller models.

Corollary 5. ***Towards Density-Optimal Training - Green Scaling Law****: The development of LLMs should shift from being performance-centric to being density-centric.*

Density is a metric that reflects the trade-off between effectiveness and efficiency. Therefore, blindly increasing model parameters to pursue performance improvements can lead to lower model density, resulting in unnecessary energy consumption. For example, while Llama-3.1-405B (Dubey et al., 2024) achieves state-of-the-art performance among open-source models, it requires computational resources that are hundreds of times greater than other models. Consequently, model developers need to shift their focus from merely optimizing performance to optimizing density. This approach aims to achieve the best results with minimal computational costs, thereby realizing a more sustainable and environmentally friendly Scaling Law.

In this work, we propose a new evaluation metric, capability density, for LLMs, which can offer a new, unified perspective on the two current trends – enhancing effectiveness and increasing efficiency. Based on our proposed metric, we evaluate 29 open-source models and find an empirical experience law, named Densing Law: the density of LLMs exhibits an exponentially increasing trend. Based on this empirical relationship, we discuss several deductions and provide observational evidence. Through this novel evaluation perspective, we hope to provide valuable insights and guidance for the future development of LLMs.

2 Density for Large Language Models

In this section, we formally define the density for LLMs, which is calculated as the ratio of the effective parameter size to the actual parameter size. In the following sections, we will first describe the overall framework and formal definition of LLM density. Then we introduce how to utilize the Scaling Law to estimate the effective parameter size.

2.1 Overall Framework and Definition

The core of LLM density lies in the effective parameter size, which refers to the number of parameters required for a reference model to achieve the same performance as a given model. To achieve this, we need to fit a function that relates the parameter sizes of the reference model to its performance. Specifically, for a given model $\mathcal{M}$ with $N_{\mathcal{M}}$ parameters, assume its performance score on the downstream tasks is $S_{\mathcal{M}}$. This score can be calculated using various metrics depending on the downstream task, such as accuracy, F1 score, etc. To compute the effective parameter size, we train a series of reference models with varying scales of parameters and training data. Based on these models, we fit a function between the parameter size and performance: $S = f(N)$, where S denotes the downstream performance, and N represents the parameter sizes of the reference model. Then we can calculate the effective parameter size as $\hat{N}(S) = f^{-1}(S)$ and the density for $\mathcal{M}$ is defined as:

$$\rho(\mathcal{M}) = \frac{\hat{N}(S_{\mathcal{M}})}{N_{\mathcal{M}}} = \frac{f^{-1}(S_{\mathcal{M}})}{N_{\mathcal{M}}}. \tag{1}$$

of the curve. Specifically, we estimate the downstream performance with the following function:

$$S = \frac{c}{1 + e^{-\gamma(\mathcal{L}-l)}} + d, \tag{3}$$

where c, γ, l, and d are parameters need to be estimated.

2.4 Density

After fitting Equation 2 and 3, given the performance $S_{\mathcal{M}}$ of a model $\mathcal{M}$, we can infer the effective parameter size by utilizing the inverse functions of these equations. It is important to note that in Equation 2, the loss $\mathcal{L}$ is a bivariate function of both the parameter count N and the training data size D. Therefore, when calculating the effective parameter size, it is necessary to specify a particular training data size D. Here, to calculate the effective parameter size, we defaultly use $D = D_0 = 1T$ tokens. Then the effective parameter size can be explained as the parameter size the reference model trained with D_0 tokens needs to achieve equivalent performance. Concretely, we can compute the effective parameter size as:

$$\hat{\mathcal{L}}(S_{\mathcal{M}}) = l - \frac{1}{\gamma} ln\left(\frac{c}{S_{\mathcal{M}} - d} - 1\right); \quad \hat{N}(S_{\mathcal{M}}) = \left(\frac{\hat{\mathcal{L}}(S_{\mathcal{M}}) - bD_0^{-\beta}}{a}\right)^{-\frac{1}{\alpha}}. \tag{4}$$

Now, we have established the relationship between the downstream performance and effective parameter size. The density of the given model $\mathcal{M}$ is $\rho(\mathcal{M}) = \frac{\hat{N}(S_{\mathcal{M}})}{N_{\mathcal{M}}}$. Intuitively, if one model can achieve better performance with the same scale of parameters, then the model's density is higher. Therefore, in the future, considering the limited computation resources of deployment devices, we should devote great effort to improving the model's density instead of merely increasing the model parameter scales for better performance.

3 Density Evolution

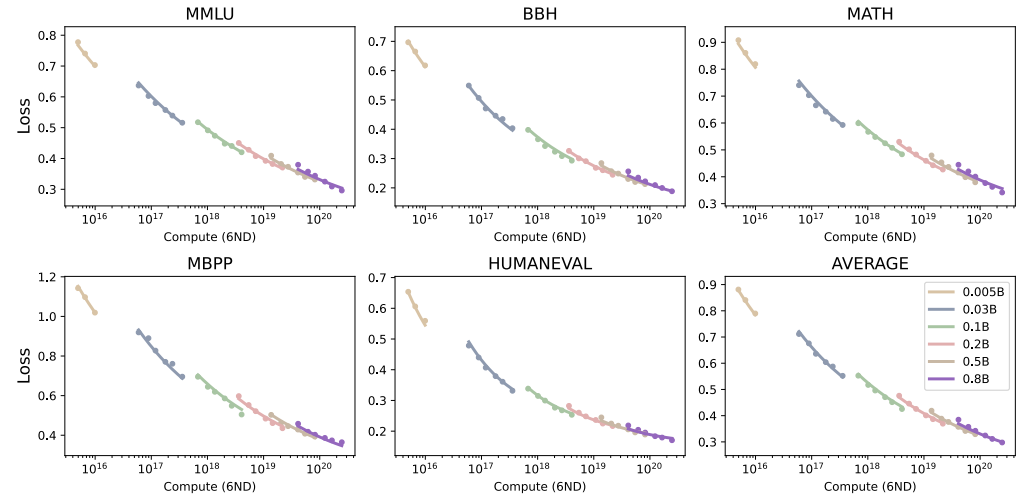
3.1 Evaluation Settings

Dataset In this work, we adopt the following widely-used datasets for evaluation: MMLU (Hendrycks et al., 2020) for English knowledge-intensive tasks, BBH (Suzgun et al., 2023) for challenging logic reasoning tasks, MATH (Hendrycks et al., 2021) for mathematical reasoning tasks, and HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021) for coding tasks. We apply the open-source tools (OpenCompass, 2023; Liu et al., 2024) for evaluation. Here, we evaluate all models in a few-shot in-context learning manner and these models are required to generate the final answer label based on the given demonstrations and inputs of test instances. Following widely-used settings, MMLU, BBH, MATH, HumanEval, and MBPP are evaluated under the 5-shot, 3-shot, 4-shot, 0-shot, and 3-shot settings, respectively. Besides, for BBH, MATH, and MBPP, we adopt the chain-of-thought prompting technique (Wei et al., 2022b).

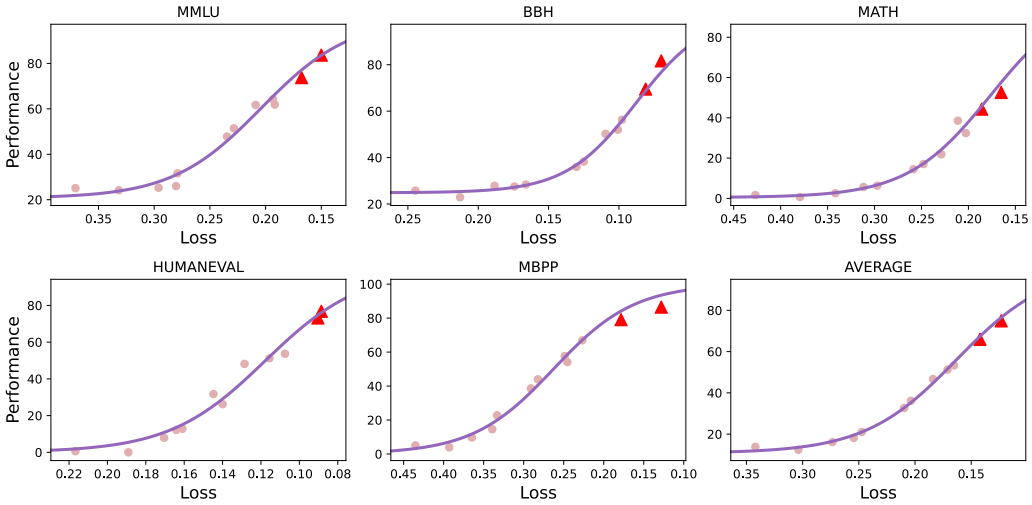
Loss Estimation Models In the loss estimation step, we need to run a series of models with different scales of parameters and training data. These models will be used as the reference models for further density computation. In this work, we adopt the training corpus of MiniCPM-3-4B (Hu et al., 2024),

Table 1: The detailed hyper-parameters of small models trained for loss estimation.

Name	# Para	BS	n_{layer}	d	d_{ffn}	d_{head}	n_{head}	n_{kv}
0.005B	5,247,232	32	8	256	640	64	4	1
0.03B	31,470,080	32	12	512	1,280	64	8	2
0.1B	106,196,736	64	18	768	1,920	64	12	3
0.2B	245,416,960	128	24	1,024	2,560	64	16	2
0.4B	476,852,480	256	30	1,280	3,200	64	20	2
0.8B	828,225,024	512	36	1,536	3,840	64	24	3



(a) Loss Estimation



(b) Performance Estimation

Figure 2: The results for loss estimation and performance estimation. Here, the lines are fitted curves. X-axis in (a) refers to the pre-training compute, which is approximated by Compute $= 6ND$. Triangles in (b) are larger models for prediction.

a widely-used edge-size model, to train the small models. As for the model architecture, we use grouped query attention (Ainslie et al., 2023), gated feedforward layers with SiLU as the activation function. We train the models using Warmup-Stable-Decay learning rate scheduler. To estimate the scaling curve, we train the models with $\{10, 15, 20, 30, 40, 60\} \times N$ tokens, where N refers to the parameter size. We list the hyper-parameters for small scaling models in Table 1.

Performance Estimation Models In the performance estimation step, we introduce additional well-trained models to fit the loss-performance curve. Specifically, we use a series well-trained MiniCPM-3 models and their intermediate training checkpoints. Their parameter scales range from 0.5 billion to tens of billion. These models use the same vocabulary as our scaling models with different parameter sizes and training datasets.

Evaluated Models Furthermore, to illustrate the change in density over time, we select widely used LLMs for evaluation since the release of Llama-1 (Touvron et al., 2023a), as most open-source models released before Llama-1 cannot achieve meaningful performance on our selected datasets. Specifically, we evaluate the density of the following models: Llama series of models (Touvron et al., 2023a,b; Dubey et al., 2024), Falcon (Almazrouei et al., 2023), MPT (Team, 2023), Phi