Are More LM Calls All You Need? Towards the Scaling Properties of Compound AI Systems

Lingjiao Chen[1], **Jared Davis**[1], **Boris Hanin**[3], **Peter Bailis**[1],
Ion Stoica[2], **Matei Zaharia**[2], **James Zou**[1]
Stanford University[1]
University of California, Berkeley[2]
Princeton University[3]

Abstract

Many recent state-of-the-art results in language tasks were achieved using compound systems that perform multiple Language Model (LM) calls and aggregate their responses. However, there is little understanding of how the number of LM calls – *e.g.,* when asking the LM to answer each question multiple times and taking a majority vote – affects such a compound system's performance. In this paper, we initiate the study of scaling properties of compound inference systems. We analyze, theoretically and empirically, how the number of LM calls affects the performance of Vote and Filter-Vote, two of the simplest compound system designs, which aggregate LM responses via majority voting, optionally applying LM filters. We find, surprisingly, that across multiple language tasks, the performance of both Vote and Filter-Vote can first increase but then decrease as a function of the number of LM calls. Our theoretical results suggest that this non-monotonicity is due to the diversity of query difficulties within a task: more LM calls lead to higher performance on "easy" queries, but lower performance on "hard" queries, and non-monotone behavior can emerge when a task contains both types of queries. This insight then allows us to compute, from a small number of samples, the number of LM calls that maximizes system performance, and define an analytical scaling model for both systems. Experiments show that our scaling model can accurately predict the performance of Vote and Filter-Vote systems and thus find the optimal number of LM calls to make.

1 Introduction

Compound AI systems that perform multiple Language Model (LM) calls and aggregate their responses are increasingly leveraged to solve language tasks [ZKC+24, DLT+23, TAB+23, TWL+24, WWS+22]. For example, Google's Gemini Ultra achieved state-of-the-art results on MMLU using a CoT@32 voting strategy: the LM is called 32 times, and then the majority vote of the 32 responses is used in the final response [TAB+23]. Other compound systems filter responses using an LM before selecting one [Alp23].

A natural question is, thus, how does scaling the number of LM calls affect the performance of such compound systems? This question is under-explored in research, but characterizing the scaling dynamics is crucial for researchers and practitioners to estimate how many LM calls are needed for their applications and allocate computational resources aptly. Understanding these scaling dynamics is also helpful in recognizing the limits of compound inference strategies.

As a first step towards answering this question, we study the scaling properties of two popular compound system designs, Vote and Filter-Vote. Vote aggregates multiple proposed answers via majority vote, as in Gemini's CoT@32. Filter-Vote leverages a filter before performing a majority

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-1455

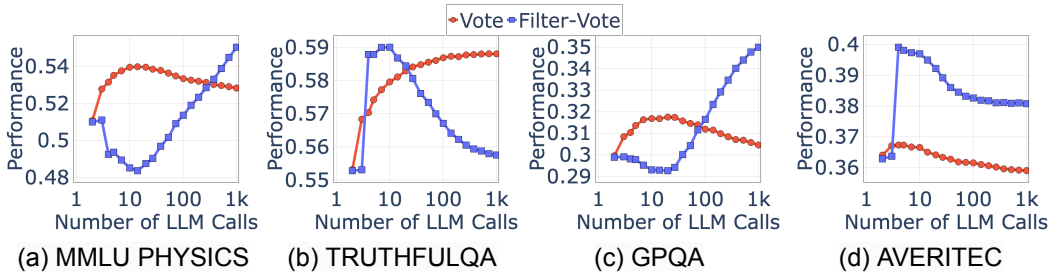


Figure 1: The scaling behavior of Vote and Filter-Vote. Interestingly, their performance is often a non-monotonic function of the number of LM calls. For example, as the number LM calls increases, Vote's performance initially increases but then decreases, while Filter-Vote's performance initially decreases but then increases on the MMLU PHYSICS dataset.

vote, similar to AlphaCode 2 [Alp23]. While the inference system design space is broad, we focus on Vote and Filter-Vote for two reasons. First, they have already been used in real applications, such as Gemini's CoT@32 strategy. Second, despite their simplicity, they exhibit nontrivial scaling properties. Specifically, although one might expect their performance to monotonically increase as more LM calls are invoked, we have identified a surprising phenomenon, across multiple language tasks, exhibited by these systems: growing the number of LM calls initially improves performance but then degrades it, as shown in Figure 1.

This surprising effect motivates our theoretical study of Vote and Filter-Vote, which explains this non-monotone effect through the diversity of *query difficulty* in a given task (see Figure 2 and Theorem 2). At a high level, our results show that more LM calls continuously lead to better performance on "easy" queries and worse performance on "hard" queries. When a task is some mixture of "easy" and "hard" queries, the non-monotone aggregate behavior emerges. Formally, a query is easy if a compound system with infinitely many LM calls gives a correct answer and hard otherwise. We provide mathematical conditions under which a query is easy or difficult for Vote or Filter-Vote. We also derive a performance scaling model for both systems that explicitly models query difficulties, and present an algorithm that lets users fit the scaling law's parameters using a small number of samples. In experiments with GPT-3.5, we show that these algorithms can let us estimate the scaling behavior for various problems and identify the optimal number of calls to make to the LM to maximize accuracy. Our main contributions are:

Main Contributions

Non-monotonic Scaling Behavior. We find empirically that the performance of Vote and Filter-Vote is often non-monotonic in the number of LM calls.

Query Difficulty-based Explanation. We propose a formal notion of query difficulty. We then argue empirically and show in a simple analytical model that the diversity of query difficulty can explain the scaling behavior of Vote and Filter-Vote. Additional LM calls improve performance on easy queries and degrades performance on difficult queries. We provide precise conditions under which the non-monotone scaling behavior emerges on datasets containing both easy and difficult queries in our analytical model.

Heuristic for Optimal Number of LM Calls. We empirically validate predictions for the optimal number of LM calls from our analytical model for Vote and Filter-Vote, which can be estimated from a small number of queries.

In a nutshell, our work shows that more LM calls do not necessarily improve the performance of compound AI systems and that it is possible, at least in some cases, to predict how the number of LM calls affects AI systems' performance and thus decide the optimal number of LM calls for a given task. Our work focuses on tasks with a fairly small number of possible responses (e.g., multiple-choice questions) that support a majority vote, but tasks with many valid outputs, such as

https://doi.org/10.52202/079017-1455

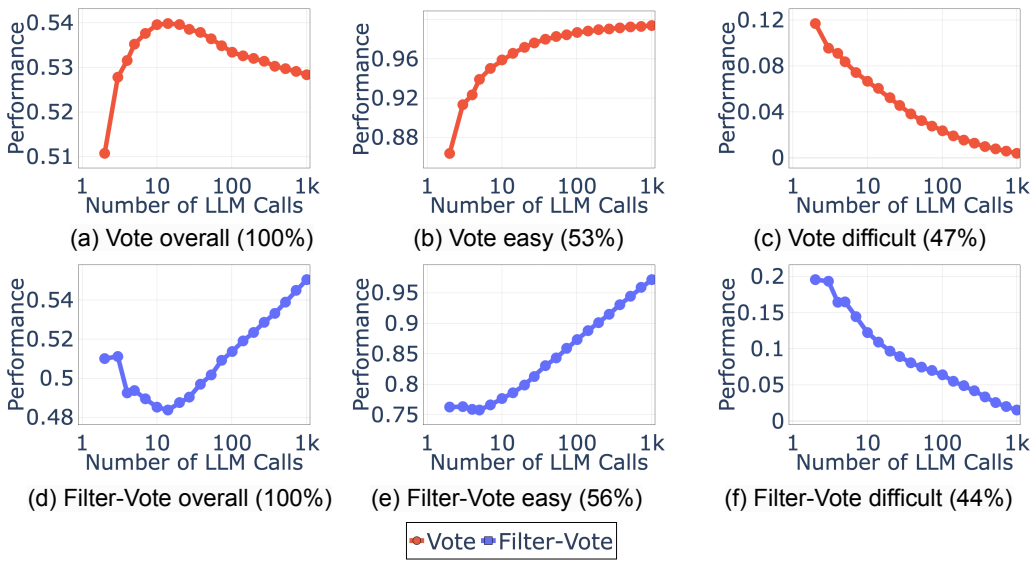


Figure 2: Performance breakdown on MMLU PYHSICS. As the number of LM calls increases, Vote and Filter-Vote perform increasingly better on easy queries but increasingly worse on difficult ones.

chat, remain under-explored. We have released the code and datasets[1] used in this paper, and hope to excite more research regarding the scaling properties of compound AI systems.

2 Related Work

Neural Scaling Laws. There has been extensive research on how the training parameters affect the performance of neural language models [KMH+20, SGS+22, BDK+21, MLP+23, IPH+24]. Among others, the model loss has been empirically shown to follow a power law as the number of model parameters and training tokens [KMH+20]. Researchers have also proposed theories to explain the empirical scaling laws [BDK+21]. By contrast, recent work has found that scaling up the model parameters leads to performance decay on certain tasks [MLP+23]. To the best of our knowledge, there is no study on how the number of LM calls affects the performance of a compound system, which is complementary to scaling laws on training parameters and data set sizes.

Compound Systems using LMs. Many inference strategies that perform multiple model calls have been developed to advance performance on various language processing tasks [XCG+23, DLT+23, TAB+23, TWL+24, WWS+22, CZZ23, ZKAW23, ŠPW23]. For example, Gemini reaches state-of-the-art performance on MMLU via its CoT@32 majority voting scheme [TAB+23]. Self-consistency [WWS+22] boosts the performance of chain-of-thought via a majority vote scheme over multiple reasoning paths generated by PaLM-540B. Finally, AlphaCode 2 [Alp23] matches the 85th percentile of humans in a coding contest regime by generating up to one million samples per problem via an LM and then filtering the answer set down. While these approaches are empirically compelling, there has been little systematic study of how the number of LM calls affects these systems' performance, and how it should be tuned for a given application.

Compound System Evaluation. Compound AI systems are increasingly evaluated in traditional benchmarks [CKB+21, HBK+21, TVCM18, NWD+19] as well as domain-specific new datasets [KBGA24, MMA+24, KCM+23] and the agent environments [LYZ+23, SYC+20, JYW+24]. We refer the interested readers to a survey for more details [CWW+24]. Existing papers focus on obtaining a single metric of a given system, while our goal is to understand how the a compound system's performance is affected by its parameters (e.g., # of LM calls).

[1] https://github.com/lchen001/CompoundAIScalingLaws

https://doi.org/10.52202/079017-1455

Algorithm 1: Vote.

Input: A user query x, # gen responses K
Output: A response $\hat{y}$

Sample $\theta_1, \theta_2, \cdots, \theta_K$ i.i.d. from Θ;
Generate $z_k = G(x, \theta_k), k = 1, \cdots, K$;
Set $\hat{y}_K = V(z_1, \cdots, z_K)$;
Return $\hat{y}_K$

Algorithm 2: Filter-Vote.

Input: A user query x, # gen responses K
Output: A response $\hat{y}$

Sample $\theta_1, \theta_2, \cdots, \theta_K$ i.i.d. from Θ;
Generate $z_k, e_k = G(x, \theta_k), k = 1, \cdots, K$;
Generate $w_k = \Phi(x, z_k, e_k), k = 1, \cdots, K$;
if $\max_k w_k = 0$ **then**
  Set $\hat{y}_K = V(z_1, \cdots, z_K)$;
else
  Set $\hat{y}_K = V(\{\!|z_k \mid w_k = 1|\!\})$;
Return $\hat{y}_K$

3 Inference System Designs

In this paper, we focus on two simple and natural inference system designs: Vote and Filter-Vote. Vote is inspired by and resembles several real-world compound AI systems, such as self-consistency [WWS+22], Medprompt [NKM+23], and Gemini CoT@32 strategy [TAB+23], while Filter-Vote represent many other real-world compound AI systems including AlphaCode 2 [Alp23] and AlphaGeometry [TWL+24]. Note that this paper focuses on tasks with a small number of possible answers.

Building Blocks. Vote and Filter-Vote rely on three building blocks, a generator $G(\cdot, \cdot)$, a majority voter $V(\cdot)$, and a filter $\Phi(\cdot, \cdot, \cdot)$. The generator $G(\cdot, \cdot)$ takes a user query x and $\theta \in \Theta$ as inputs and produces a candidate answer and an explanation. Here, instantiations of Θ are a design choice of users and can encode many generation strategies. For example, even with a single fixed LM, diverse generations may be achieved by using a non-zero temperature and different prompt wordings or few-shot examples for each call to the LM. If Θ contains different LMs, then this system definition can also represent LM ensembles. The majority voter V returns the mode of its input, i.e., $V(z_1, z_2, \cdots, z_K) \triangleq \arg\max_{a \in A} \sum_{k=1}^{K} \mathbb{1}_{z_k = a}$, and breaks ties arbitrarily. Here, A is the space of all possible answers. Finally, the filter $\Phi(\cdot, \cdot)$ takes the user query and multiple candidate answers as input, and only returns the subset that an LM believes is correct.

Vote. Given a user query x, Vote (i) first creates K candidate answers by calling the generator G, and then (ii) uses the majority voter V to choose one as the final response $\hat{y}_K$. The details are given in Algorithm 1.

Filter-Vote. Given a user query, Filter-Vote (i) first generates multiple candidate answers, (ii) removes a few candidate answers by the filter Φ, and (iii) then uses the majority voter V to choose one from the remaining answers as the final response. If all answers are removed by the filter, then V is applied on the original candidate answers. Algorithm 2 gives the formal description.

4 Analytical Model of Scaling Behavior

Now we present our analytical performance model of Vote and Filter-Vote strategies. Specifically, we are interested in understanding the behavior of $F(K; D) \triangleq \mathbb{E}[\hat{y}_K = y]$, where the expectation is over D and the candidate responses. All notations are summarized in Table 1.

4.1 When do more LM calls lead to an increase or decrease in performance?

Our first key insight is that individual query difficulty is crucial in LM calls' effects. To see this, let us first introduce query difficulty indicator.

Definition 1. *Given a user query x, $d(x)$ is called an query difficulty indicator if*

$$\lim_{K \to \infty} F(K, x) = \begin{cases} 0 & \textit{iff } d(x) > 0, \\ 1 & \textit{iff } d(x) < 0 \end{cases}$$

Table 1: Notations.

Symbol	Meaning
x	an input query
y	the correct answer
K	the number of LM calls
z_k	the output by one LM call
$\hat{y}_K$	the output by an inference system using K LM calls
D/D_{Tr}	test dataset/train dataset
A	answer space
α	fraction of easy queries
p_1	probability of z_k being correct for easy queries
p_2	probability of z_k being correct for difficult queries
$F(K;D)$	Accuracy of an Inference System with K LM calls per query on D
$G(K;D)$	Analytical Performance Model (to approximate $F(K;D)$)

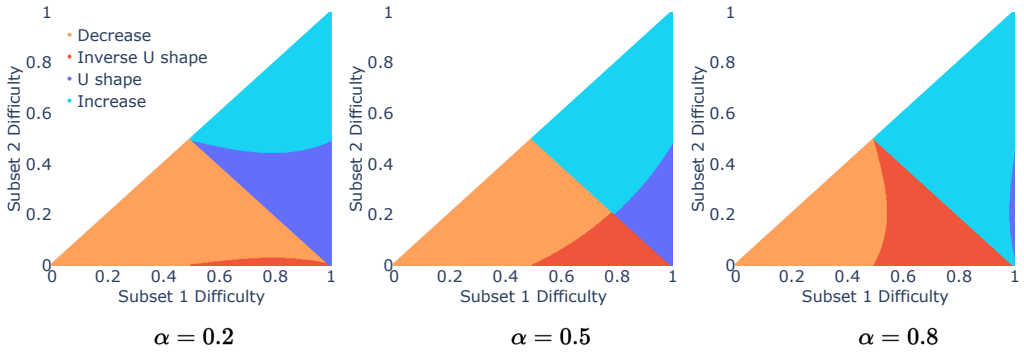


Figure 3: How the query difficulties shape the landscape of a one-layer Voting Inference System's performance. Informally, if the overall task is "easy" ($p_1 + p_2 > 1$), but the fraction of "hard" queries is large ($\alpha < 1 - \frac{1}{t}$), then as the number of LM calls increases, the Voting Inference Systems' performance increases first but then decreases. We call such a landscape a "inverse U shape". Similarly, if the overall task is "hard" ($p_1 + p_2 < 1$), but the fraction of "hard" queries is small ($\alpha > 1 - \frac{1}{t}$), then enlarging the number of LM calls leads an initial decrease and then increase. Such a landscape is called a "U shape". When α is large, the U-shape is less likely to occur while the inverse U-shape becomes more common. Smaller α leads to an opposite trend.

Intuitively, a positive query difficulty indicator implies the query is difficult, i.e., infinitely many LM calls lead to an incorrect final answer, and a negative value implies the query is easy, i.e., infinitely many LM calls eventually give a correct final answer. For simplicity, we assume that the limit of $F(K, x)$ is always either 0 or 1 for Vote and Filter-Vote, i.e., eventually the answer is correct or incorrect. Also note that $d(x)$ is scale-invariant, i.e., if $d(x)$ is an item difficulty indicator, then for any positive scalar $\gamma > 0$, $\gamma \times d(x)$ is also a difficulty indicator. We will call a query x difficult (easy) if $d(x) > 0$ ($d(x) < 0$). We give two concrete instantiations as follows.

Lemma 1. *For Vote,* $d_V(x) \triangleq \max_{a \neq y} \Pr[G(x,\theta) = a] - \Pr[G(x,\theta) = y]$ *is an query difficulty indicator. For Filter-Vote, denote* $G(x,\theta) = [G_1(x,\theta), G_2(x,\theta)]$. *Then* $d_F(x) \triangleq \max_{a \neq y} \Pr[G_1(x,\theta) = a | \Phi(x, G_1(x,\theta), G_2(x,\theta)) = 1] - \Pr[G_1(x,\theta) = y | \Phi(x, G_1(x,\theta), G_2(x,\theta)) = 1]$ *is a query difficulty indicator.*

https://doi.org/10.52202/079017-1455

Here, $d_V(x) > 0$ indicates that an incorrect answer is more likely to be generated than the correct one, hence the query is difficult. Similarly, $d_F(x) > 0$ implies that an incorrect answer is more likely to be kept in the filtered answer set.

More LM calls elicit higher performance on easy queries, but lower performance on difficult queries. Therefore, the performance, $F(K; D)$, is more difficult to characterize when the data set D contains both easy and difficult queries. Here, we study Vote on a special case of D to understand how the difficulty impacts the performance function $F(K; D)$.

A case study on a specific dataset. Let us consider a specific dataset D_{α,p_1,p_2} with answer space cardinality $|A| = 2$. Here, $\alpha \in [0, 1]$ queries in D are x_1 such that $Pr[G(x_1, \theta) = y] = p_1 > \frac{1}{2}$, and $1 - \alpha$ queries are x_2 such that $Pr[G(x_2, \theta) = y] = p_2 < \frac{1}{2}$. The following theorem qualitatively characterizes the performance of Vote on this dataset.

Theorem 2. *Let* $t \triangleq \frac{p_2(1-p_2)(\frac{1}{2}-p_2)}{p_1(1-p_1)(p_1-\frac{1}{2})} + 1$. *If* $p_1 + p_2 \neq 1$ *and* K *is odd, then* $F(K; D_{\alpha,p_1,p_2})$

- *increases monotonically, if* $p_1 + p_2 > 1$ *and* $\alpha \geq 1 - \frac{1}{t}$
- *decreases monotonically, if* $p_1 + p_2 < 1$ *or* $\alpha \leq 1 - \frac{1}{t}$
- *increases and then decreases, if* $p_1 + p_2 > 1$ *and* $\alpha < 1 - \frac{1}{t}$
- *decreases and then increases, if* $p_1 + p_2 < 1$ *and* $\alpha > 1 - \frac{1}{t}$

Theorem 2 precisely connects the query difficulty with the performance landscape. Here, t is a constant that only depends on p_1 and p_2, i.e., the probability of an LM's generation being correct on easy and hard queries, respectively. Intuitively, t quantifies the difficulty similarity between the easy and hard queries: it becomes larger if the easy queries are more difficult (p_1 is smaller) or the hard queries are less difficult (p_2 is larger). Interestingly, it suggests that, for some query difficulty distribution, a non-monotone effect of the number of LM calls is expected. Informally, if the overall task is "easy" ($p_1 + p_2 > 1$), but the fraction of "hard" queries is large ($\alpha < 1 - \frac{1}{t}$), then as the number of LM calls increases, the Voting Inference Systems' performance increases first but then decreases. We call such a landscape a "inverse U shape". Similarly, if the overall task is "hard" ($p_1 + p_2 < 1$), but the fraction of "hard" queries is small ($\alpha > 1 - \frac{1}{t}$), then enlarging the number of LM calls leads an initial decrease and then increase. Such a landscape is called a "U shape". This well explains the U-shape of Inference Systems' performance shown in Figure 1.Figure 3 visualizes the effects of query difficulty on the performance landscape in more detail.

4.2 What is the analytical scaling model?

Now we derive an analytical scaling model for both Vote and Filter-Vote. Noting that the performance is the average of $F(K, x)$ for each x in the dataset, the key challenge is identifying the shape of $F(K, x)$ for easy and difficult queries. Let us first consider the special case $|A| = 2$, where we can obtain a close form result for Vote.

Theorem 3. *If* $|A| = 2$, *then on any query* x, *the performance of Vote is* $F(K, x) = I_{\frac{1-d_V(x)}{2}}(\frac{K+1}{2}, \frac{K+1}{2})$, *where* $I_x(a, b) \triangleq \int_0^x t^{a-1}(1-t)^{b-1}dt / \int_0^1 t^{a-1}(1-t)^{b-1}dt$ *is the regularized incomplete beta function.*

Thus, for Vote with $|A| = 2$, $F(K, D) = \mathbb{E}_{x\sim D}[I_{\frac{1-d_V(x)}{2}}(\frac{K+1}{2}, \frac{K+1}{2})]$.

How about Filter-Vote and the general answer space? Admittedly, an exact scaling model is challenging to obtain. Instead, we give an approximation model inspired by the special case. We first note that $F(K, x)$ should be treated separately for difficult and easy queries: after all, as a function of a, the incomplete beta function $I_x(a, a)$ monotonically increases/decreases if $x > \frac{1}{2}$ ($x < \frac{1}{2}$). Second, $I_x(a, a)$ grows roughly exponentially in x, and this trend should hold for general answer space and for both Vote and Filter-Vote. Hence, we propose the following scaling model

$$G(K, x) \triangleq \begin{cases} e^{-c_1(x)K - c_2(x)\sqrt{K} + c_3(x)}, & \text{if } d(x) > 0, \\ 1 - e^{-c_1(x)K - c_2(x)\sqrt{K} + c_3(x)}, & \text{if } d(x) < 0 \end{cases}$$

where constants $c_1(x) > 0, c_2(x) > 0, c_3(x)$ do not depend on the number of LM calls K. Therefore, our analytical performance scaling model is $G(K, D) = \mathbb{E}_{x\sim D}[G(K, x)]$. In practice, one can use a training dataset D_{Tr} to fit the parameters in $G(K, D)$. Note that given a query x, the parameters $c_i(x)$ can be different for Vote and Filter-Vote. In particular, if the filter is of high quality, then the performance should converge quickly, and thus the constants $c_i(\cdot)$ are likely to be larger. Otherwise, the performance should scale slower, and thus the constants $c_i(\cdot)$ should be smaller. We will show in the experiments that $G(K, D)$ matches the empirical performance $F(K, D)$ accurately.

4.3 How to optimize the number of LM calls?

In general, one can always (i) fit the analytical scaling model $G(K, D)$, and (ii) then use $\max_K G(K, D)$ to obtain the optimal number of LM calls. Interestingly, we show that for a special case, we can derive the optimal number of LM calls.

Theorem 4. *If $p_1 + p_2 > 1$ and $\alpha < 1 - \frac{1}{t}$, then the number of LM calls K^* that maximizes $F(K, D_{\alpha,p_1,p_2})$ for Vote (up to rounding) is*

$$K^* = 2\frac{\log \frac{\alpha}{1-\alpha}\frac{2p_1-1}{1-2p_2}}{\log \frac{p_2(1-p_2)}{p_1(1-p_1)}}$$

The optimal number of LM calls depends on the query difficulty. For example, K^* will be larger if α grows (up to $1 - \frac{1}{t}$). That is, if there are more "easy" queries than "difficult" queries, then more LM calls should be adopted.

5 Experiments

We compare the empirical performance of Vote and Filter-Vote and the performance predicted by our analytical scaling model. Our goal is three-fold: (i) validate that there are cases where more LM calls do not monotonically improve the performance of these Inference Systems, (ii) justify that the number of LM calls has opposite effects on easy and difficult queries, and (iii) explore whether our analytical scaling model can accurately predict the performance of Vote and Filter-Vote, and thus guide the design of Inference Systems such as optimizing number of LM calls.

Datasets and LM. To understand the scaling properties of Vote and Filter-Vote, we conduct systematical experiments on both (i) real-world datasets and (ii) synthetic datasets with controlled query difficulties. Specifically, the real-world datasets include MMLU PHYSICS [HBB+20], TRUTHFULQA [LHE21], GPQA [RHS+23], and AVERITEC [SGV24]. MMLU PHYSICS contains high school physics questions extracted from the original MMLU dataset. TRUTHFULQA measures whether a language model is truthful in generating answers to questions. GPQA queries are generated by experts in biology, physics, and chemistry. Each query in AVERITEC is a claim and the goal is to verify its correctness. Each query is prompted as a multiple-choice question for objective evaluation. The details of these datasets and prompts can be found in the Appendix. The synthetic dataset is D_{α,p_1,p_2} as introduced in Section 3, and we study the scaling behavior by varying the parameters. We use GPT-3.5-turbo-0125 on the real-world datasets. All experiments are averaged over 1,000 runs.

Non-monotonic Scaling Behavior. We start by understanding how the number of LM calls affects the performance of Vote and Filter-Vote empirically. As shown in Figure 1, we observe a non-monotonic behavior: more LM calls can sometimes lead to a drop in performance! This underscores the importance of scaling performance modeling.

A case study on AVERITEC. Now let us perform a case study on the AVERITEC dataset to understand the intriguing behavior better. In particular, we use the deployment partition of AVERITEC [SGV24], which contains 500 fact verification questions. The goal is to determine if a given claim should be (A) refused, (B) supported, or there is (C) conflicting evidence or (D) not enough evidence. We evaluate the performance of both Vote and Filter-Vote on AVERITEC. In addition, we fit our analytical scaling model with 2, 5, 10, 20, 50, 100 LM calls. Then we use it to predict the performance of Vote and Filter-Vote using 100 randomly drawn number of LM calls.

 https://doi.org/10.52202/079017-1455

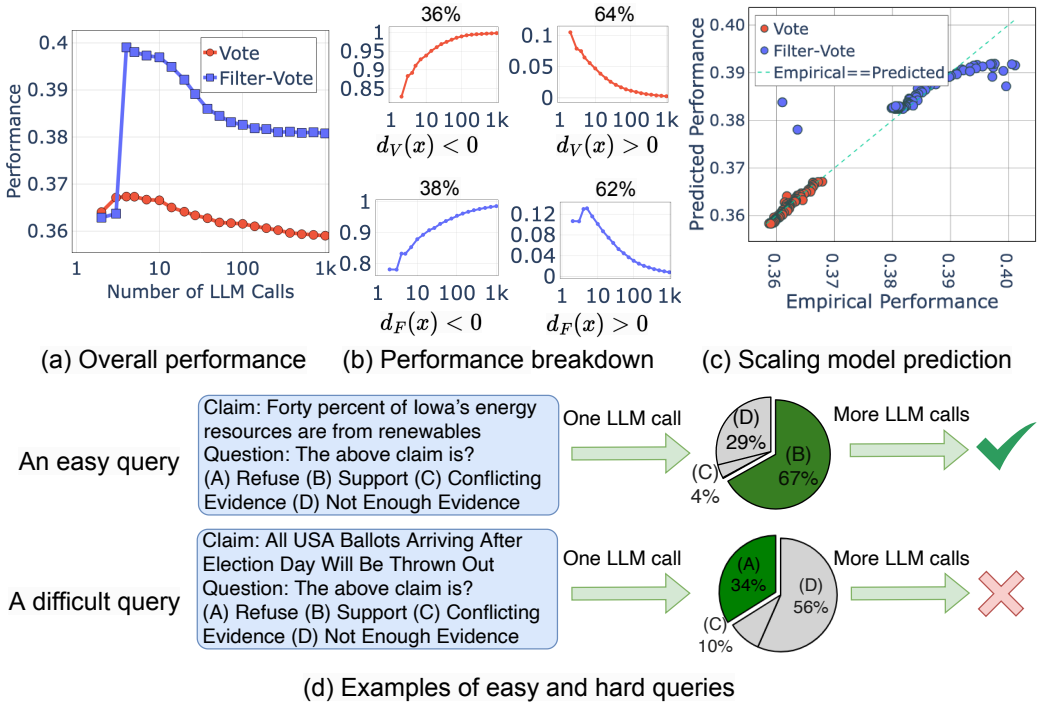
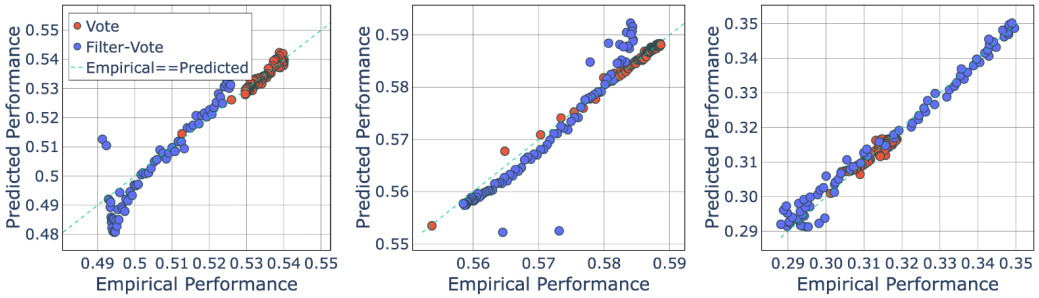


Figure 4: A case study on the AVERITEC dataset. (a) As more LM calls are invoked, the overall performance of Vote and Filter-Vote both initially increases but then decreases. (b) This U-shape can be perfected explained by the opposite effects on easy and difficult queries: More LM calls lead to higher performance on easy queries, but lower performance on difficult ones. (c) Our analytical scaling model accurately predicts the empirical performance. (d) Examples of an easy query and a difficult one. One LM call gives the correct answer with probability higher than any other answers (67%), and thus Vote with more calls eventually gives the correct answer. For the difficult query, the probability of the correct answer (34%) is lower than that of an incorrect answer (56%). Thus, Vote with more LM calls eventually always generates a wrong answer.

As shown in Figure 4, there are several intriguing behaviors. First, more LM calls do not always lead to better performance. In fact, the performance of both Vote and Filter-Vote increases first but then decreases as the number of LM calls increases from 2 to 1000 (Figure 4(a)). The performance breakdown (shown in Figure 4(b)) gives a natural explanation. More LM calls lead to higher performance when a query is relatively difficult ($d_F(x) > 0$ for Filter-Vote and $d_V(x) > 0$ for Vote), but lower performance when a query is relatively easy ($d_F(x) < 0$ for Filter-Vote and $d_V(x) < 0$ for Vote). We also observe a high correlation between the performance predicted by our proposed analytical scaling model and the empirical performance, as depicted in Figure 4 (c). This implies the optimal number of LM calls can also be accurately predicted by our scaling model. Finally, we also give examples of one easy query and one difficult query in Figures 4 (d). On the easy example, one LM call gives the correct answer with a probability higher than any other answer (67%). Thus, more LM calls eventually give the correct answer. On the difficult query, the probability of the correct answer (34%) is lower than that of an incorrect answer (56%). Thus, Vote with more LM calls eventually always generates a wrong answer.

Scaling model performance on real-world datasets. Next we study how our analytical scaling model generalizes to other real-world datasets. In particular, we fit our analytical model with 2, 5, 10, 50, and 100 LM calls, and then use the scaling model to predict the performance on 100 randomly drawn number of LM calls.

Figure 5 shows the correlation between the predicted and empirical performance on three real-world datasets, namely, MMLU PHYSICS, TRUTHFULQA, and GPQA. We first note that the best performance of the Filter-Vote system is not necessarily better than that of the Vote system. Indeed,



(a) MMLU PYHSICS (c) TRUTHFULQA (b) GPQA

Figure 5: How the proposed analytical scaling model performs on other real-world datasets, namely, (a) MMLU PHYSICS, (b) TRUTHFULQA, and (c) GPQA. Here, we fit the scaling model with 2, 5, 10, 50, and 100 LM calls, and then use the scaling model to predict the performance on 100 randomly drawn number of LM calls. Overall, we observe that our analytical model can predict the performance of the two compound systems accurately. Interestingly, the prediction accuracy on the Vote system is relatively higher than that on the Filter-Vote system. This is expected as the Filter-Vote system generalizes the Vote system and thus its scaling behavior can be more complex.

on TRUTHFULQA, the Vote system's best performance is higher than that of the Filter-Vote system. This further highlights the importance of performance prediction, even if there is no budget constraint. Second, we observe that our scaling model can predict the performance of both Vote and Filter-Vote systems accurately across all these real-world datasets. This is because our scaling model carefully takes into account both difficult and easy queries and thus reflects the non-monotone behavior. Interestingly, the prediction accuracy on the Vote system is higher than that on the Filter-Vote system. This is perhaps because the Filter-Vote system generalizes the Vote system and thus its scaling behavior can be more complex.

Difficulty distribution determines whether more LM calls help. To systematically understand how the query difficulty affects Inference Systems' performance landscape, we synthesize a bi-level difficult dataset D_{α,p_1,p_2}, vary (i) the fraction of the easy subset α and (ii) the query difficulty p_1, p_2, and then study the scaling performance of Vote. When it is clear from the context, we may also call (p_1, p_2) query difficulty.

As shown in Figure 6, we observe that query difficulty plays an important role in the number of LM calls' effects. For example, when the difficulty parameter is $(0.85, 0.1)$ and the fraction of easy queries is $\alpha = 0.6$, Vote's performance is monotonically increasing as the number of LM calls grows. However, adding more hard queries by changing the fraction $\alpha = 0.4$ changes the trend: the performance goes down first for small call numbers and then goes up for larger numbers of calls. It is also interesting to notice an inverse "U"-shape. For example, when query difficulty is $(0.85, 0.4)$, there is a clear U-shape performance. Overall, this justifies that (i) there are cases where more LM calls is not beneficial, and (ii) the diversity of query difficulty critically determines these cases.

Analytical scaling model predicts the optimal number of LM calls. Identifying the optimal number of calls is an important implication of the scaling properties. Here, we compare the optimal number of calls for Vote predicted by our analytical model and the optimal numbers empirically observed on the bi-level difficult dataset D_{α,p_1,p_2}. As summarized in Table 2 (in the Appendix), the predicted optimal number is exactly the observed optimal number of LM calls, for all query difficulties evaluated. This validates the assumptions made by Theorem 4.

6 Conclusion

In this paper, we systematically study how the number of LM calls affects the performance of two natural inference strategy designs: majority voting (Vote) and majority voting after filtering results with an LM (Filter-Vote). We find that increasing the number of LM calls can lead to non-monotone behavior, *e.g.,* first increasing performance and then decreasing it. We offer theoretical analysis

https://doi.org/10.52202/079017-1455

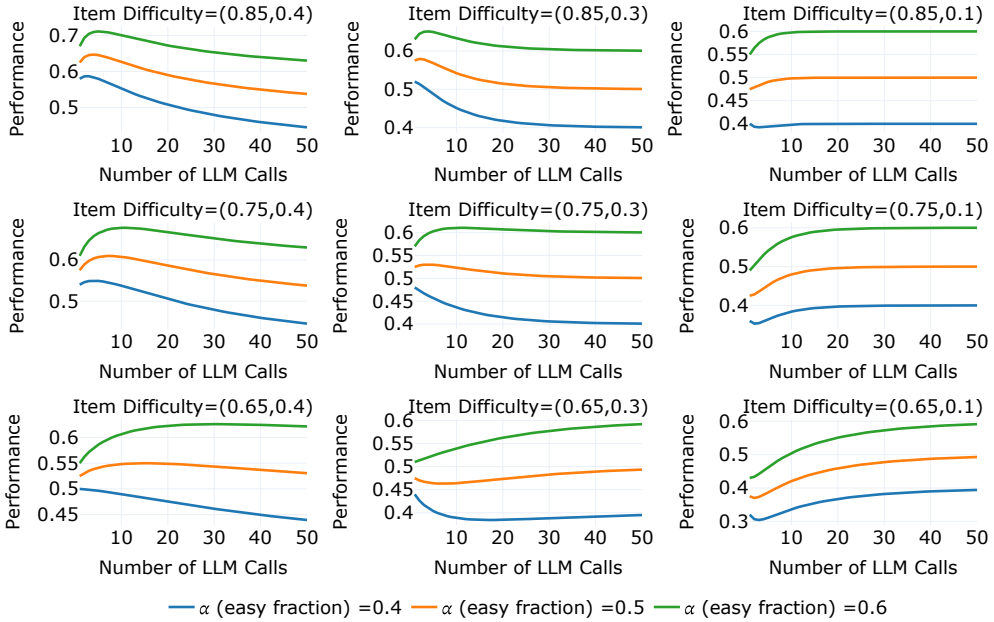


Figure 6: Overall performance of one-layer Voting Inference Systems on synthetic datasets with bi-level difficulty. Overall, we observe that increasing the number of LM calls does not necessarily lead to performance improvements. For example, when the query difficulty is $(0.85, 0.4)$, the performance increases first and then decreases as the number of LM calls grows. When the query difficulty is $(0.65, 0.1)$, there is a reverse trend: the performance goes down first and then goes up. This validates our empirical observation that a larger number of LM calls and thus more resources may not necessarily result in better performance.

that attributes this phenomenon to the diversity of query difficulties within a task, and conduct experiments to validate our analysis. Furthermore, we show how to estimate the optimal number of LM calls to make for a given task using a small number of queries to fit the parameters of our analytical model, thus helping practitioners optimize their system designs. Overall, our study shows that more LM calls are not necessarily better and underscores the importance of compound system design. To stimulate further research, we have released our code and datasets, available at `https://github.com/lchen001/CompoundAIScalingLaws`. We hope our findings and analysis will inspire more research into maximizing the effectiveness of inference time compute.

Acknowledgement

This work was supported in part by a Google PhD Fellowship, a Sloan Fellowship, NSF CCF 1763191, NSF CAREER AWARD 1651570 and 1942926, NIH P30AG059307, NIH U01MH098953, grants from the Chan-Zuckerberg Initiative, Sutherland, and affiliate members and other supporters of the Stanford DAWN project and UC Berkeley SKY Lab, including Google, IBM, Intel, Lacework, Meta, Microsoft, Mohamed Bin Zayed University of Artificial Intelligence, Nexla, Samsung SDS, Uber, and VMware. BH is supported by a 2024 Sloan Fellowship in Mathematics, NSF CAREER grant DMS-2143754, and NSF grants DMS-1855684 and DMS-2133806. We also thank anonymous reviewers for helpful discussion and feedback.

References

[Alp23] AlphaCode. Alphacode 2 technical report. *https://storage.googleapis.com/deepmind-media/AlphaCode2/AlphaCode2_Tech_Report.pdf*, 2023.

[BDK+21] Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *arXiv preprint arXiv:2102.06701*, 2021.

[CKB+21] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[CWW+24] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.

[CZZ23] Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

[DLT+23] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

[HBB+20] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[HBK+21] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[IPH+24] Berivan Isik, Natalia Ponomareva, Hussein Hazimeh, Dimitris Paparas, Sergei Vassilvitskii, and Sanmi Koyejo. Scaling laws for downstream task performance of large language models. *arXiv preprint arXiv:2402.04177*, 2024.

[JYW+24] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024.

[KBGA24] Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo. Gpt-4 passes the bar exam. *Philosophical Transactions of the Royal Society A*, 382(2270):20230254, 2024.

[KCM+23] Tiffany H Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, et al. Performance of chatgpt on usmle: potential for ai-assisted medical education using large language models. *PLoS digital health*, 2(2):e0000198, 2023.

[KMH+20] 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*, 2020.

[LHE21] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

[LYZ+23] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*, 2023.

[MLP+23] Ian R McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, et al. Inverse scaling: When bigger isn't better. *arXiv preprint arXiv:2306.09479*, 2023.

 https://doi.org/10.52202/079017-1455

[MMA+24] Nikita Mehandru, Brenda Y Miao, Eduardo Rodriguez Almaraz, Madhumita Sushil, Atul J Butte, and Ahmed Alaa. Evaluating large language models as agents in the clinic. *NPJ digital medicine*, 7(1):84, 2024.

[NKM+23] Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. Capabilities of gpt-4 on medical challenge problems. *arXiv preprint arXiv:2303.13375*, 2023.

[NWD+19] Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*, 2019.

[RHS+23] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

[SGS+22] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

[SGV24] Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. Averitec: A dataset for real-world claim verification with evidence from the web. *Advances in Neural Information Processing Systems*, 36, 2024.

[ŠPW23] Marija Šakota, Maxime Peyrard, and Robert West. Fly-swat or cannon? cost-effective language model choice via meta-modeling. *arXiv preprint arXiv:2308.06077*, 2023.

[SYC+20] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

[TAB+23] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[TVCM18] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*, 2018.

[TWL+24] Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

[WWS+22] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[XCG+23] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023.

[ZKAW23] Jieyu Zhang, Ranjay Krishna, Ahmed H Awadallah, and Chi Wang. Ecoassistant: Using llm assistant more affordably and accurately. *arXiv preprint arXiv:2310.03046*, 2023.

[ZKC+24] Matei Zaharia, Omar Khattab, Lingjiao Chen, Jared Quincy Davis, Heather Miller, Chris Potts, James Zou, Michael Carbin, Jonathan Frankle, Naveen Rao, and Ali Ghodsi. The shift from models to compound ai systems. `https://bair.berkeley.edu/blog/2024/02/18/compound-ai-systems/`, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim an interesting scaling behavior of simple compound AI systems and propose a possible explanation by query difficult diversity, which are both justified by theoretical analysis and empirical results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please see Section B in the appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

https://doi.org/10.52202/079017-1455

Answer: [Yes]

Justification: All proofs can be found in Section C in the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all experiment details in Section 5 and Section D in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will release the data and code once the paper is accepted.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so No is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please see Section 5 and Section D in the appendix for details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our experiments are conducted on large-scale datasets (a few hundred to a thousand samples).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

https://doi.org/10.52202/079017-1455

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer to Section 5 and Section D for details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read and followed the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please refer to Section A in the Appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No new dataset is released or collected, and thus safeguards are not necessary.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited all datasets used in this paper and also explicitly included their licenses in Section D in the appendix.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

https://doi.org/10.52202/079017-1455

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new data is released in this paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not use crowdsourcing or human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

A Broader Impacts

Many inference strategies are often computationally and financially expensive to deploy due to multiple calls of language models. Our study suggests not blindly scaling up, but determining the resources allocated to these inference strategies carefully. More broadly, our paper paves the way for data-centric inference strategy designs.

B Limitations

In this paper, we focus our analysis and experiments on the scaling behaviors for two instances of compound AI systems, but there are many other types of compound AI systems. For ease of evaluation, our experiments are conducted on relatively objective language tasks, but it remains open to study how the performance scales on subjective tasks such as generating poems and writing essays. Another interesting open problem is how to predict query difficult without querying the LMs. Note that in this work, we do not discuss the cost of LM calls; this is an important dimension to weigh in practice, and in future work we will investigate how to balance cost, performance, and latency scaling.

C Missing Proofs

We give all missing proofs here.

C.1 Useful Lemmas

Lemma 5. *Suppose D is 2-level difficult with (α, p_1, p_2) and $\|A\| = 2$, and W.L.O.G, K is odd. Then $F(K; D) = \alpha I_{p_1}(\frac{K+1}{2}, \frac{K+1}{2}) + (1-\alpha) I_{p_2}(\frac{K+1}{2}, \frac{K+1}{2})$, where $I_x(a,b) \triangleq \int_0^x t^{a-1}(1-t)^{b-1}dt / \int_0^1 t^{a-1}(1-t)^{b-1}dt$ is the regularized incomplete beta function.*

Proof. To analyze how the performance scales as the network size increases, let us first expand the performance of Inference System by the law of total expectation

$$F(K; D) = \mathbb{E}[\hat{y} = y] = \sum_{i=1}^{2} \mathbb{E}[\hat{y} = y | r(x) = p_i] \Pr[r(x) = p_i] \tag{1}$$

Now consider $\mathbb{E}[\hat{y} = y | r(x) = p]$ for any p. Note that there are in total 2 possible generations (by $\|A\| = 2$), so the estimated generation $\hat{y}$ is correct if and only if more than half of the generations is correct. That is to say, $\mathbb{E}[\hat{y} = y | r(x) = p] = \Pr[\sum_{k=1}^{K} \mathbb{1}_{z_k = y} > \frac{K}{2}] = 1 - \Pr[\sum_{k=1}^{K} \mathbb{1}_{z_k = y} \leq \frac{K}{2}]$. For ease of notation, denote $Z \triangleq \sum_{k=1}^{K} \mathbb{1}_{z_k = y}$. Note that $\mathbb{1}_{z_k = y}$ is a Bernoulli variable with parameter p and thus Z is a Binomial variable with parameter (K, p). Therefore, we have $Pr[Z \leq w] = I_{1-p}(K - w, w + 1)$, where $I_x(a,b) \triangleq \int_0^x t^{a-1}(1-t)^{b-1}dt$ is the incomplete beta function. Thus, the above gives us

$$\mathbb{E}[\hat{y} = y | r(x) = p] = 1 - I_{1-p}(K - \lfloor \frac{K}{2} \rfloor, \lfloor \frac{K}{2} \rfloor + 1) = I_p(\lfloor \frac{K}{2} \rfloor + 1, K - \lfloor \frac{K}{2} \rfloor)$$

If K is odd, we can write it as $\mathbb{E}[\hat{y} = y | r(x) = p] = I_p(\frac{K+1}{2}, \frac{K+1}{2})$ We can now plug this back into equation (1) to obtain

$$\begin{aligned} F(K; D) &= \sum_{i=1}^{2} \mathbb{E}[\hat{y} = y | r(x) = p_i] \Pr[r(x) = p_i] = \sum_{i=1}^{2} I_{p_i}(\frac{K+1}{2}, \frac{K+1}{2}) \Pr[r(x) = p_i] \\ &= \alpha I_{p_1}(\frac{K+1}{2}, \frac{K+1}{2}) + (1-\alpha) I_{p_2}(\frac{K+1}{2}, \frac{K+1}{2}) \end{aligned}$$

which completes the proof. □

Lemma 6. *$I_p(M+1, M+1) = I_p(M, M) + \frac{p^M (1-p)^M}{M B(M,M)}(2p-1)$, where $B(a,b) \triangleq \int_0^1 t^{a-1}(1-t)^{b-1}dt$ is the beta function.*

https://doi.org/10.52202/079017-1455

Proof. Noting that $I_p(a+1,b) = I_p(a,b) - \frac{p^a(1-p)^b}{aB(a,b)}$, we have

$$I_p(M+1,M+1) = I_p(M,M+1) - \frac{p^M(1-p)^{M+1}}{MB(M,M+1)}$$

By $I_p(a,b+1) = I_p(a,b) + \frac{p^a(1-p)^b}{bB(a,b)}$, we have

$$I_p(M,M+1) = I_p(M,M) + \frac{p^M(1-p)^M}{MB(M,M)}$$

Thus, $I_p(M+1,M+1) = \frac{p^M(1-p)^M}{MB(M,M)} - \frac{p^M(1-p)^{M+1}}{MB(M,M+1)}$. The Pascal's identity implies that $B(a,b+1) = \frac{a}{a+b}B(a,b)$ and thus $B(M,M+1) = B(M,M)/2$. Thus, we have

$$I_p(M+1,M+1) = \frac{p^M(1-p)^M}{MB(M,M)} - \frac{p^M(1-p)^{M+1}}{MB(M,M+1)} = \frac{p^M(1-p)^M}{MB(M,M)} - \frac{2p^M(1-p)^{M+1}}{MB(M,M)}$$

Extracting the common factors gives

$$\frac{p^M(1-p)^M}{MB(M,M)} - \frac{2p^M(1-p)^{M+1}}{MB(M,M)} = \frac{p^M(1-p)^M}{MB(M,M)}[1-2(1-p)] = \frac{p^M(1-p)^M}{MB(M,M)}(2p-1)$$

That is, $I_p(M+1,M+1) = I_p(M,M) + \frac{p^M(1-p)^M}{MB(M,M)}(2p-1)$, which completes the proof. □

C.2 Proof of Lemma 1

Proof. Let us first consider Vote.

We want to first note that $\lim_{K\to\infty}\Pr[\hat{y}_K = y^*] = 1$, where $y^* \triangleq \arg\max_{a\in A}\Pr[G(x,\theta)=a]$. To see this, we can first write $\Pr[\hat{y}_K = y^*] = \Pr[\max_{a\neq y^*}\sum_{i=1}^K \mathbb{1}_{z_i=a} < \sum_{i=1}^K \mathbb{1}_{z_i=y^*}] = \Pr[\sum_{i=1}^K \mathbb{1}_{z_i=a} - \sum_{i=1}^K \mathbb{1}_{z_i=y^*} < 0, \forall a \neq y^*]$. Subtracting 1 from both sides, we have $1-\Pr[\hat{y}_K = y^*] = \Pr[\cup_{a\neq y^*}\sum_{i=1}^K \mathbb{1}_{z_i=a} - \sum_{i=1}^K \mathbb{1}_{z_i=y^*} > 0]$. Now by union bound, we can bound the right hand side by $\sum_{a\neq y}\Pr[\sum_{i=1}^K \mathbb{1}_{z_i=a} - \mathbb{1}_{z_i=y^*} > 0]$. Now, note that each term within the inner summation is i.i.d. Furthermore, observe that the expectation is $\mathbb{E}[\mathbb{1}_{z_i=a} - \mathbb{1}_{z_i=y^*}] = \Pr[G(x,\theta)=a] - \Pr[G(x,\theta)=y^*] < 0$. Therefore, by law of large numbers, we have $\lim_K \Pr[\sum_{i=1}^K \mathbb{1}_{z_i=a} - \mathbb{1}_{z_i=y^*} > 0] = 0$. Since there are only finite number of $a \in A$, the sum of this quantity over all a should also be 0. Therefore, we have

$$\lim_{K\to\infty} 1 - \Pr[\hat{y}_K = y^*] = \lim_{K\to\infty}\Pr[\cup_{a\neq y^*}\sum_{i=1}^K \mathbb{1}_{z_i=a} - \sum_{i=1}^K \mathbb{1}_{z_i=y^*} > 0] = 0$$

Thus, $\lim_{K\to\infty}\Pr[\hat{y}_K = y^*] = 1$.

Now, $d_V(x) > 0$ implies $y = y^*$ and thus $\lim_{K\to\infty} F(K,D) = 1$. $d_V(x) > 0$ implies $y \neq y^*$ and thus $\lim_{K\to\infty} F(K,D) = 0$.

Now let us consider Filter-Vote.

Abusing the notation a little, we claim that again $\lim_{K\to\infty}\Pr[\hat{y}_K = y^*] = 1$, where $y^* \triangleq \arg\max_{a\in A}\Pr[G_1(x,\theta) = a | \Phi(x, G_1(x,\theta), G_2(x,\theta)) = 1]$, where recall that $G_1(x,\theta), G_2(x,\theta) \triangleq G(x,\theta)$. To see this, we can first write $\Pr[\hat{y}_K = y^*] = \Pr[\max_{a\neq y^*}\sum_{i=1}^K \mathbb{1}_{z_i=a} w_i < \sum_{i=1}^K \mathbb{1}_{z_i=y^*} w_i] = \Pr[\sum_{i=1}^K \mathbb{1}_{z_i=a} w_i - \sum_{i=1}^K \mathbb{1}_{z_i=y^*} w_i < 0, \forall a \neq y^*]$. Subtracting 1 from both sides, we have $1 - \Pr[\hat{y}_K = y^*] = \Pr[\cup_{a\neq y^*}\sum_{i=1}^K \mathbb{1}_{z_i=a} w_i - \sum_{i=1}^K \mathbb{1}_{z_i=y^*} w_i > 0]$. Now by union bound, we can bound the right hand side by $\sum_{a\neq y}\Pr[\sum_{i=1}^K \mathbb{1}_{z_i=a} w_i - \mathbb{1}_{z_i=y^*} w_i > 0]$. Now, note that each term within the inner summation is i.i.d. Furthermore, observe that the expectation is $\mathbb{E}[\mathbb{1}_{z_i=a} w_i - \mathbb{1}_{z_i=y^*} w_i] = \Pr[G_1(x,\theta) = a, \Phi(x, G_1(x,\theta), G_2(x,\theta)) = 1] - \Pr[G(x,\theta) = y^*, \Phi(x, G_1(x,\theta), G_2(x,\theta)) = 1] < 0$. Therefore,

by law of large numbers, we have $\lim_K \Pr[\sum_{i=1}^K \mathbb{1}_{z_i=a} w_i - \mathbb{1}_{z_i=y^*} w_i > 0] = 0$. Since there are only finite number of $a \in A$, the sum of this quantity over all a should also be 0. Therefore, we have

$$\lim_{K\to\infty} 1 - \Pr[\hat{y}_K = y^*] = \lim_{K\to\infty} \Pr[\cup_{a\neq y^*} \sum_{i=1}^K \mathbb{1}_{z_i=a} w_i - \mathbb{1}_{z_i=y^*} w_i > 0] = 0$$

Thus, $\lim_{K\to\infty} \Pr[\hat{y}_K = y^*] = 1$.

Now, $d_F(x) > 0$ implies $y = y^*$ and thus $\lim_{K\to\infty} F(K, D) = 1$. $d_F(x) > 0$ implies $y \neq y^*$ and thus $\lim_{K\to\infty} F(K, D) = 0$.

Hence, we have shown that $d_V(x)$ and $d_F(x)$ are difficulty indicator for Vote and Filter-Vote, respectively. □

C.3 Proof of Theorem 2

Proof. We construct a recurrent relation on $F(K; D)$ by Applying Lemma 5

$$\begin{aligned}
&F(K+2;D) - F(K;D)\\
=&\alpha I_{p_1}(\frac{K+3}{2}, \frac{K+3}{2}) + (1-\alpha) I_{p_2}(\frac{K+3}{2}, \frac{K+3}{2}) - [\alpha I_{p_1}(\frac{K+1}{2}, \frac{K+1}{2}) + (1-\alpha) I_{p_2}(\frac{K+1}{2}, \frac{K+1}{2})]\\
=&\alpha(I_{p_1}(\frac{K+3}{2}, \frac{K+3}{2}) - I_{p_1}(\frac{K+1}{2}, \frac{K+1}{2})) + (1-\alpha)(I_{p_2}(\frac{K+3}{2}, \frac{K+3}{2}) - I_{p_2}(\frac{K+1}{2}, \frac{K+1}{2}))
\end{aligned}$$

For ease of notation, let us denote $M = \frac{K+1}{2}$. Applying Lemma 6 leads to

$$\begin{aligned}
&F(K+2;D) - F(K;D)\\
=&\alpha[\frac{p_1^M(1-p_1)^M}{MB(M,M)}(2p_1-1)] + (1-\alpha)[\frac{p_2^M(1-p_2)^M}{MB(M,M)}(2p_2-1)]\\
=&\frac{1}{MB(M,M)}[\alpha \cdot p_1^M(1-p_1)^M(2p_1-1) + (1-\alpha)\cdot p_2^M(1-p_2)^M(2p_2-1)]
\end{aligned}$$

Now rearranging the terms gives

$$\begin{aligned}
&F(K+2;D) - F(K;D)\\
=&\frac{1}{MB(M,M)} \cdot (1-\alpha) p_2^M(1-p_2)^M(1-2p_2) \cdot [\frac{\alpha(2p_1-1)}{(1-\alpha)(1-2p_2)}[\frac{p_1(1-p_1)}{p_2(1-p_2)}]^M - 1]
\end{aligned}$$

For ease of notation, denote $\Delta F(M) \triangleq \frac{\alpha(2p_1-1)}{(1-\alpha)(1-2p_2)}\frac{p_1(1-p_1)}{p_2(1-p_2)}]^M - 1$. Note that $\Delta F(M)$ is monotonically increasing or decreasing, depending on the parameters p_1, p_2. It is also easy to show that $\alpha \geq 1 - \frac{1}{t}$ if and only if $\Delta F(1) \geq 0$. Now, we are ready to derive the main results by analyzing $\Delta F(M)$.

- $p_1 + p_2 > 1$ and $\alpha \geq 1 - \frac{1}{t}$: Now $\frac{p_1(1-p_1)}{p_2(1-p_2)} > 1$, and thus $\Delta F(M)$ is monotonically increasing. Furthermore, it is easy to see $\lim_{M\to\infty} \Delta F(M) = \infty$. Furthermore, $\Delta F(1) \geq 0$. That is, for any given M, $\Delta F(M)$ is non-negative and thus $F(K; D)$ must be monotonically increasing

- $p_1 + p_2 > 1$ and $\alpha < 1 - \frac{1}{t}$: Now $\frac{p_1(1-p_1)}{p_2(1-p_2)} > 1$, and thus $\Delta F(M)$ is monotonically increasing. Furthermore, it is easy to see $\lim_{M\to\infty} \Delta F(M) = \infty$. Furthermore, $\Delta F(1) < 0$. That is, as K and thus M increases, $\Delta F(M)$ is negative and then becomes positive. Therefore, $F(K; D)$ must decrease and then increase

- $p_1 + p_2 < 1$ and $\alpha > 1 - \frac{1}{t}$: Now $\frac{p_1(1-p_1)}{p_2(1-p_2)} < 1$, and thus $\Delta F(M)$ is monotonically decreasing. Furthermore, it is easy to see $\lim_{M\to\infty} \Delta F(M) = -1$. Furthermore, $\Delta F(1) > 0$. That is, as K and thus M increases, $\Delta F(M)$ is positive and then becomes negative. Therefore, $F(K; D)$ must increase and then decrease

https://doi.org/10.52202/079017-1455

- $p_1+p_2 < 1$ and $\alpha \leq 1-\frac{1}{t}$: Now $\frac{p_1(1-p_1)}{p_2(1-p_2)} < 1$, and thus $\Delta F(M)$ is monotonically decreasing. Furthermore, it is easy to see $\lim_{M\to\infty} \Delta F(M) = -1$. Furthermore, $\Delta F(1) \leq 0$. That is, for any given M, $\Delta F(M)$ is non-positive and thus $F(K; D)$ must be monotonically decreasing

Thus, we complete the proof. □

C.4 Proof of Theorem 3

Proof. Let us first consider $F(K, x) = \Pr[\hat{y}_K = y]$. Note that there are in total 2 possible generations (by $\|A\| = 2$), so the estimated generation $\hat{y}$ is correct if and only if more than half of the generations is correct. That is to say, $\mathbb{E}[\hat{y}_K = y] = \Pr[\sum_{k=1}^{K} \mathbb{1}_{z_k=y} > \frac{K}{2}] = 1 - \Pr[\sum_{k=1}^{K} \mathbb{1}_{z_k=y} \leq \frac{K}{2}]$. For ease of notation, denote $Z \triangleq \sum_{k=1}^{K} \mathbb{1}_{z_k=y}$. Note that $\mathbb{1}_{z_k=y}$ is a Bernoulli variable with parameter p and thus Z is a Binomial variable with parameter (K, p). Therefore, we have $Pr[Z \leq w] = I_{1-p}(K-w, w+1)$, where $I_x(a, b) \triangleq \int_0^x t^{a-1}(1-t)^{b-1}dt / \int_0^1 t^{a-1}(1-t)^{b-1}dt$ is the incomplete beta function. Specifically, we can set $w = \frac{K}{2}$ and get $F(K, x) = \Pr[\hat{y}_K = y] = I_{1-p}(\frac{K+1}{2}, \frac{K+1}{2})$. Noting that by definition, $d_V(x) = 1 - 2p$, we can rewrite this as $F(K, x) = I_{\frac{1-d_V(x)}{2}}(\frac{K+1}{2}, \frac{K+1}{2})$. Taking the integral over x completes the proof. □

C.5 Proof of Theorem 4

Proof. Note that the optimal number of LM calls must ensure that the incremental component $\Delta F(M) = 0$. That is, the optimal value M^* must satisfy $\Delta F(M^*) = \frac{\alpha(2p_1-1)}{(1-\alpha)(1-2p_2)}[\frac{p_1(1-p_1)}{p_2(1-p_2)}]^{M^*} - 1 = 0$. Solving the above equation gives us a unique solution

$$M^* = 2\frac{\log \frac{\alpha}{1-\alpha}\frac{2p_1-1}{1-2p_2}}{\log \frac{p_2(1-p_2)}{p_1(1-p_1)}}$$

Noting that $K^* = \lceil 2M \rceil]$ completes the proof. □

D Additional Experiment Details

D.1 Datasets

We evaluate Vote and Filter-Vote on four real-world datasets, namely, GPQA [RHS+23], AVERITEC [SGV24], TRUTHFULQA [LHE21], and MMLU PHYSICS [HBB+20]. AVERITEC is a fact verification dataset, where each query is a claim (e.g., "All USA ballots Arriving after election day will be thrown out"), and the goal is to determine if this claim should be refused, supported, or there is conflicting evidence or not enough confidence. We use the "development" partition offered in the original paper, which contains 500 queries. GPQA is a dataset consisting of multiple-choice questions written by domain experts in biology, physics, and chemistry. We use the diamond partition, which contains 198 queries, and as reported by the original paper, enjoys the highest expert accuracy. TRUTHFULQA is another widely used multiple-choice question-answering dataset. We uses the validation parititon, which contains 817 questions spanning 38 categories, including health, law, finance, and politics. MMLU PHYSICS is the high school physics subset of the MMLU dataset, which contains 151 questions. All queries are prompted as multiple-choice questions for ease of evaluation.

Liscense. GPQA is released under the MIT License. AVERITEC uses Creative Commons Attribution-NonCommercial 4.0 International License. TRUSTFULQA and PHYSICS (as part of MMLU) both adopt the Apache License 2.0

D.2 Prompting for generators and filters

We provide the prompts used for the generators and filters in the following boxes.

Table 2: Optimal number of LM calls prediction. For each data distribution with specific 2-level difficulty parameters α, p_1, p_2, we sample 100 data points, employ a simulated Voting Inference System with 1000 number of LM calls, estimate the parameters by their empirical mean and adopt the analytical optimal number of LM calls predictor. Across evaluated query difficulties, our predicted optimal number of LM calls exactly matches the optimal number empirically observed.

α	p_1	p_2	$\hat{\alpha}$	$\hat{p}_1$	$\hat{p}_2$	Optimal Number of LM Calls
						Analytical/Empirical
0.4	0.85	0.4	0.42	0.84	0.40	3
0.4	0.85	0.3	0.38	0.85	0.30	1
0.4	0.75	0.4	0.41	0.74	0.40	4
0.4	0.75	0.3	0.44	0.75	0.30	1
0.4	0.65	0.4	0.41	0.65	0.40	1
0.5	0.85	0.4	0.50	0.84	0.40	4
0.5	0.85	0.3	0.50	0.85	0.30	2
0.5	0.75	0.4	0.51	0.75	0.40	7
0.5	0.75	0.3	0.49	0.75	0.30	4
0.5	0.65	0.4	0.48	0.65	0.40	15
0.6	0.85	0.4	0.59	0.84	0.39	5
0.6	0.85	0.3	0.63	0.85	0.30	4
0.6	0.75	0.4	0.61	0.75	0.40	11
0.6	0.75	0.3	0.59	0.75	0.30	11
0.6	0.65	0.4	0.62	0.65	0.40	30

Prompt for the generator $G()$

Please answer the following question. You should first analyze it step by step. Then generate your final answer by the answer is (X).
Q: {Query}
A:

Prompt for the filter $\Phi()$

[User Question]: {query}
[Answer]:{answer}
Instruction: Review your previous answer and find problems with your answer. Finally, conclude with either [[correct]] if the above answer is correct or [[wrong]] if it is incorrect. Think step by step.
Verdict:

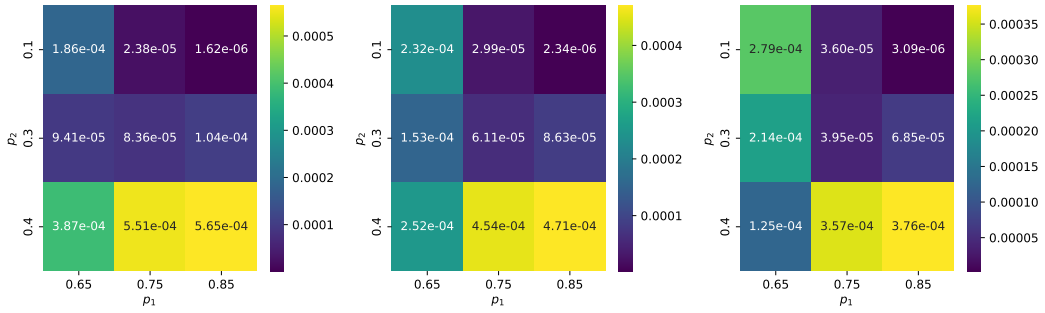
D.3 Generation Setup

We set up the temperature as 0.1 since all the evaluation tasks are objective and have clear correct answers. For each question, we query GPT-3.5-turbo-0125 400 times. Then for each $F(K, D)$, we randomly sample K answers from the 400 answers with replace to simulate the performance once, and report the average over 1000 runs.

D.4 The optimal number of LM calls predicted by the scaling model

Table 2 compares the optimal number of LM calls predicted by the scaling model and the ground-truth value on the bi-level dataset D_{α,p_1,p_2}. Overall, We observe that they match very well.

https://doi.org/10.52202/079017-1455



(a) $\alpha = 0.4$ (b) $\alpha = 0.5$ (c) $\alpha = 0.6$

Figure 7: Mean square error of the performance predicted by our proposed scaling law on synthesized datasets with varying bi-level difficulties. Here, we fit the scaling law by performance evaluated at $K = 1, 2, 3, 4, 5$, and evaluate its performance for K from 1 to 100. Overall, we observe that the predicted performance accurately matches the empirical evaluation.

D.5 Performance Estimation via Our Proposed Scaling Law

Can our proposed scaling model predict the empirical performance of Vote accurately? To answer this, we apply it to each dataset considered in Figure 6. In particular, we feed our estimator with the Inference Systems' performance with the number of LM calls being $1, 2, 3, 4, 5$, and then use it to predict the performance for LM calls within the range from 1 to 100. Note that this is quite challenging, as the estimator needs to extrapolate, i.e., predict the performance of a Voting Inference System whose number of LM calls is much larger than any number seen in the training data.

As shown in Figure 7, the performance predicted by our estimator accurately matches the empirical observation. Across all query difficulty parameters, the mean square error ranges from $1e-6$ to $1e-4$. This suggests that predicting how the number of LM calls affects a Voting Inference System is feasible and also indicates that our scaling law captures the key performance trend effectively.