SCIENCE CHINA
Information Sciences

• LETTER •

August 2025, Vol. 68, Iss. 8, 189101:1–189101:2
https://doi.org/10.1007/s11432-024-4442-8

The impact of task similarity on performance drift of LLMs: a theoretical and empirical analysis

Xuanming NI, Qiaochu ZHAO, Song HUANG* & Lian YU

School of Software and Microelectronics, Peking University, Beijing 102600, China

Received 18 June 2024/Revised 25 January 2025/Accepted 14 May 2025/Published online 3 July 2025

Citation Ni X M, Zhao Q C, Huang S, et al. The impact of task similarity on performance drift of LLMs: a theoretical and empirical analysis. Sci China Inf Sci, 2025, 68(8): 189101, https://doi.org/10.1007/s11432-024-4442-8

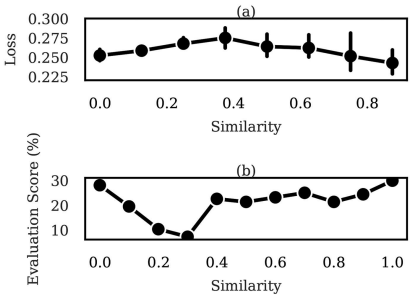
Performance drift refers to the phenomenon that large language models (LLMs), after being fine-tuned for specific tasks, may exhibit a decline in performance on other tasks. This phenomenon has been observed in recent iterations of state-of-the-art LLMs, notably GPT-3.5 and GPT-4 [1]. A critical question emerges from this phenomenon: can an LLM's capabilities be uniformly enhanced across all tasks through the exclusive use of fine-tuning techniques? This study addresses this question by associating the phenomenon of performance drift with catastrophic forgetting in continual learning. We note that in previous investigations of catastrophic forgetting, it has been observed and examined that intermediate task similarity leads to the most severe forgetting [2, 3]. While widely acknowledged within the field of continual learning, there is a paucity of research on the relationship between task similarity and performance drift in the context of LLMs. To address this research gap, this study presents both theoretical and empirical analyses of the impact of task similarity on performance drift in the context of LLMs. In our theoretical study, we employ a simplified model of causal language models, as established by previous studies [4, 5], which reduces the task of predicting the next token to predicting binary labels. Additionally, we view the distribution of natural language data as mixtures of subpopulations. By employing these methods, our results apply to arbitrary algorithms without the necessity of strict assumptions such as convergence or optimality. Our results suggest that the average loss of any arbitrary algorithm on a data subpopulation can be lower bounded by an expression maximized when the similarity between this data subpopulation and another subpopulation in the training set approaches 1/2. In our empirical study, we validated our theoretical results on both synthetic data and real-world data. Our experimental findings also suggest that intermediate task similarity has the most detrimental effect on an LLM's performance.

In our theoretical analysis, we regard an LLM as a distribution over strings of tokens. Typically, an LLM is a model that outputs the probability distribution of the next possible token, conditioned on a specific input string. Ignoring the issue of computational cost, this conditional probability distribution is equivalent to the distribution of all output strings conditioned on the input. Consequently, it can be posited that an LLM is equivalent to a probability distribution over strings. To elaborate, given an LLM defined by $\Pr(y \mid z)$ where z and y are two strings serving as input and output, respectively, we have the distribution of all strings $\Pr(s) = \Pr(s \mid z = \text{ empty string})$. Conversely, given the distribution of all strings $\Pr(s)$, we have the conditional probability $\Pr(y \mid z) = \Pr(z{+}y)/\sum_{y'} \Pr(z{+}y')$. Therefore, it is reasonable to conclude that an LLM is a distribution over strings.

For the purposes of this study, it is still necessary to define the concepts of task and task similarity. By regarding an LLM as a distribution, it can be posited that tasks are subpopulations of the whole distribution. Assuming a total of N tasks, we have the whole distribution $M(x) = \sum_{i\in[N]} \alpha_i M_i(x), \sum_{i\in[N]} \alpha_i = 1$, where $[N]$ denotes the set $\{1, 2, \ldots, N\}$, x is an arbitrary string, and M_i is the ith subpopulation of M. Furthermore, we adopt a more simplified assumption made in [4, 5]. Under this assumption, each subpopulation M_i is characterized by a binary reference string c_i of fixed length d. Each sample from subpopulation M_i is drawn by truncating c_i by a random length l and then flipping each bit of it independently with probability δ. A notable benefit of this assumption is that it enables a straightforward definition of the similarity between two subpopulations by the proportion of identical bits in each subpopulation's reference string, i.e., $r(i, j) = \frac{1}{d}\sum_{k=1}^{d} \mathbf{1}(c_i(k) = c_j(k))$. Likewise, the similarity between a sample z_0 and a subpopulation M_j can be defined as $r_j(z_0) = \frac{1}{\text{len}(z_0)} \sum_{k=1}^{\text{len}(z_0)} \mathbf{1}(z_0(k) = c_j(k))$.

Henceforth, let $D(\alpha, C)$ denote the data distribution determined by $\alpha = (\alpha_i)_{i\in[N]}$, $C = (c_i)_{i\in[N]}$, Z denote a dataset of size n that follows $Z \sim D^n(\alpha, C)$, and $h \sim A(Z)$ denote the model h trained by algorithm A on Z. The focus of this study is the model's performance on each individual subpopulation. The model's error rate on the jth subpopulation given dataset Z is defined as $\overline{\text{err}}_j(c_j, A \mid Z) = \mathbb{E}_{h\sim A(Z)}\, \mathbb{E}_{z\sim M_j}\, \mathbf{1}(h(z) \neq c_j(\text{len}(z) + 1))$. And the model's error rate on the jth subpopulation given α, C, n, and A is

* Corresponding author (email: huangsong@ss.pku.edu.cn)



(c)

Prompt style	HumanEval score (%)	MATH score (%)
Question: Solve 2*c=2 for c. Solution: 1	31.71	29.85
'Solve 2*c=2 for c.' Answer: '1'	10.98	0
def solve_linear_equation(): """ Solve 2*c=2 for c. """ return (1)	14.63	9.7

Figure 1 (a) Loss on subpopulation M_0 of the GPT-2 model; (b) HumanEval scores of the fine-tuned Mistral 7B model; (c) evaluation scores of the fine-tuned Mistral 7B model. Please refer to Appendixes B and C for complete results.

defined as $\overline{\mathrm{err}}_j(\alpha, C, A) = \mathbb{E}_{Z\sim D^n(\alpha,C)}\,\overline{\mathrm{err}}_j(c_j, A \mid Z)$.

In addition to the data distribution, it is also necessary to impose constraints on the model's capacity. We posit that a sample z_0 from M_i will only impact $\overline{\mathrm{err}}_j(c_j, A \mid Z)$ when the algorithm incorrectly identifies it as belonging to M_j. The following assumptions adopt a simplified linear effect.

Assumption 1. An algorithm A will incorrectly identify a sample z_0 from subpopulation i as belonging to subpopulation j with a probability proportional to the similarity $r_j(z_0)$. That is, $p_{\mathrm{mis}} = k_{\mathrm{mis}} r_j(z_0)$.

Assumption 2. Given a specific dataset $Z = \{z_1, \ldots, z_l\} \cup \{z_0\}$, where $z_1, \ldots, z_l \sim M_j$ and $z_0 \sim M_i, i \neq j$, for every prompt z, $\mathrm{len}(z) \geqslant \mathrm{len}(z_0) - 1$, it is assumed that z_0 has no impact on h. For any prompt $z, \mathrm{len}(z) = k < \mathrm{len}(z_0) - 1$, it is assumed that if the algorithm A incorrectly identifies z_0 as belonging to M_j, the model's error rate on prompt z will decrease by a constant factor λ_l if $z_0(k+1) \neq c_j(k+1)$, and will increase by a constant factor μ_l if $z_0(k+1) = c_j(k+1)$.

Utilizing the assumptions above, we are able to provide the following theorem, which states that training data from a subpopulation of intermediate similarity harm the model's performance on the original subpopulation most.

Theorem 1. Suppose the dataset Z is drawn from the distribution $D^n(\alpha, C)$. Let A^* denote the optimal algorithm and let $\mathrm{OPT}_j(A^*) = \overline{\mathrm{err}}_j(1, c_j, A^*)$. Then the averaged error rate of any algorithm A can be lower bounded as

$$\overline{\mathrm{err}}_j(\alpha, C, A) \geqslant \mathrm{OPT}_j(A^*) + f_{n,d,\alpha,C}(r(i,j)), \qquad (1)$$

where $f_{n,d,\alpha,C}$ takes its positive maximum near $\frac{1}{2}$.

Proof sketch. For subpopulations of high similarity to M_j, the model is prone to misidentification. However, since the similarity is high, the impact on $\overline{\mathrm{err}}_j(\alpha, C, A)$ is negligible. Conversely, for subpopulations of low similarity to M_j, misidentification has a significant impact on $\overline{\mathrm{err}}_j(\alpha, C, A)$. However, since the similarity is low, the misidentification is minimal. Hence, the conclusion is drawn. Full theoretical deductions are available in Appendix A.

Experiments on synthetic data. To validate Theorem 1, we pretrained LLMs on synthetic data generated by our data assumption. Data from two subpopulations of varying similarity r, defined as M_0 and M_1, were used in the pretraining. Thereafter, the model was evaluated on a test set from M_0. We selected GPT-2 and Llama 3.1 as the base models for pretraining, and conducted the above experiments for each model 5 times to mitigate the impact of randomness. Partial results are presented in Figure 1(a). As illustrated, the model's test loss is averagely worst when the similarity r is close to 1/2, which aligns with Theorem 1.

Experiments on real-world tasks. Two different experiments on real-world tasks were conducted, employing models such as Llama 3.1 8B and Mistral 7B and evaluating datasets such as HumanEval, BoolQ, and MATH. In the first experiment, we separately fine-tuned the model on several deliberately constructed linear equation solving datasets. While the contents in these datasets are identical, we employed different prompt styles to control the datasets' similarity to HumanEval. After fine-tuning, the model was evaluated on HumanEval and MATH. Partial results are presented in Figure 1(c). The results demonstrate that neither the most similar fine-tuning dataset nor the most dissimilar fine-tuning dataset harms the model's HumanEval score most. In the second experiment, we constructed fine-tuning datasets by blending the original dataset with random token patterns. By adjusting the proportion of the blend, we controlled the similarity between the dataset and the original task. After fine-tuning, the model was evaluated on the original task. Partial results are presented in Figure 1(b). The results indicate that the model's performance is poorest with intermediate similarity.

Conclusion and future work. This study examines the relationship between task similarity and performance drift, highlighting the limitations of fine-tuning in uniformly improving an LLM's performance across diverse tasks. However, the simplified setting of our theoretical analysis may affect the persuasiveness of the results. Future work could explore more comprehensive theoretical frameworks and develop strategies to mitigate performance drift.

Supporting information Appendixes A–C. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Chen L, Zaharia M, Zou J. How is ChatGPT's behavior changing over time? Harvard Data Sci Rev, 2024. doi: 10.1162/99608f92.5317da47
2 Ramasesh V V, Dyer E, Raghu M. Anatomy of catastrophic forgetting: hidden representations and task semantics. In: Proceedings of International Conference on Learning Representations, 2021
3 Lee S, Goldt S, Saxe A. Continual learning in the teacher-student setup: impact of task similarity. In: Proceedings of the 38th International Conference on Machine Learning, 2021. 6109–6119
4 Brown G, Bun M, Feldman V, et al. When is memorization of irrelevant training data necessary for high-accuracy learning? In: Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing, 2021. 123–132
5 Kang M, Lee S, Baek J, et al. Knowledge-augmented reasoning distillation for small language models in knowledge-intensive tasks. In: Proceedings of Advances in Neural Information Processing Systems, 2023. 48573–48602