

Length and Meaning for Language Model Maps

Momose Oyama^{1,2} Ryo Kishino¹ Hiroaki Yamagiwa^{3*} Hidetoshi Shimodaira^{1,2}

¹Kyoto University ²RIKEN ³SB Intuitions

oyama.momose@sys.i.kyoto-u.ac.jp, kishino.ryo.32s@st.kyoto-u.ac.jp

hiroaki.yamagiwa@sbintuitions.co.jp, shimo@i.kyoto-u.ac.jp

Abstract

Various studies have proposed methods to quantify and compare the rapidly growing number of large language models (LLMs). However, when constructing a macroscopic map of LLMs, there is a trade-off between estimation accuracy and computational cost, making the selection of evaluation texts critical. In this work, we propose Predictive Importance Sampling (PredIS), which selects informative texts by predicting likelihood variance from text-level features. Experiments on 100 LLMs across 25 domains demonstrate that our predictor generalizes robustly. Furthermore, we reveal that text importance cannot be explained by text length alone, as it is also closely tied to meaning reflected in the embedding space.

1 Introduction

Language model maps (Oyama et al., 2025b) visualize many language models (LLMs) in a two-dimensional space by reducing the dimensionality of their log-likelihood vectors (Fig. 1). They support understanding model relationships, visualizing training trajectories, and predicting model performance (Harada et al., 2025; Kishino et al., 2026).

Since constructing a language model map involves computing log-likelihoods for a set of texts, the selection of these texts is crucial. Using more texts improves accuracy, but this increases the cost proportionally. Oyama et al. (2025b) constructed the map via uniform sampling from the Pile corpus (Gao et al., 2020), but approximately half of the sampled texts turned out to be redundant. The informativeness of texts for distinguishing models varies widely, and prioritising high-variance texts can recover the same map with roughly half the number of texts (Oyama et al., 2025a).

However, methods for reducing this inefficiency a priori have yet to be established. For example,

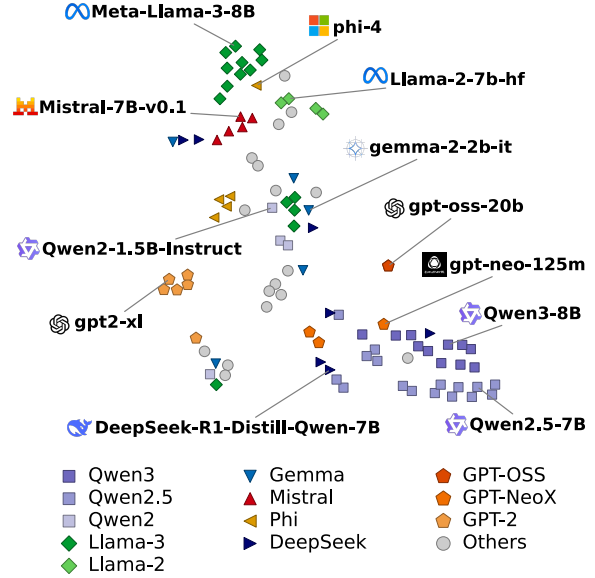


Figure 1: A map of 100 language models obtained by t-SNE visualization of log-likelihood vectors. Each point corresponds to one of 100 LLMs publicly available on the Hugging Face Hub. Colors and shapes indicate model families. The relative positions reveal relationships among LLMs. See Section 2 for details.

when remapping LLMs to different domains to analyze model relationships in each domain, we need to calculate log-likelihoods. This computational cost could be reduced if we could identify informative texts before running the target LLMs.

In this work, we focus on the remapping setting, and address this problem. We show that the likelihood variance can be well predicted from the text *length* and *meaning*, and we propose Predictive Importance Sampling (PredIS), which selects texts using a learned variance predictor.

Our contributions are summarized as follows: (i) We propose PredIS for efficient remapping. (ii) We show across 100 LLMs and 25 text domains that the predictor generalizes broadly. (iii) Analysis of PredIS reveals that text importance is captured by *meaning* beyond what *length* alone explains.

*Work done while at Kyoto University.

<https://github.com/shimo-lab/modelmap>

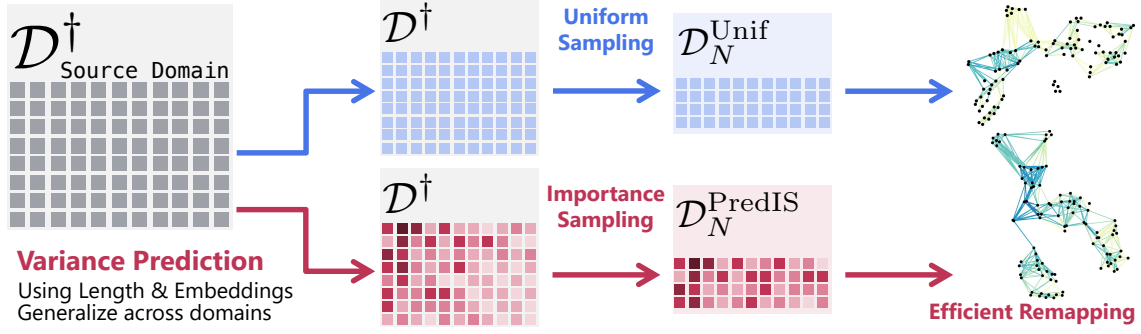


Figure 2: **Overview of PredIS.** Given a population $\mathcal{D}^\dagger$, PredIS predicts the likelihood variance of each text from its length and embedding, then samples texts for efficient remapping that generalizes across diverse domains.

2 Background

Language model maps. Given a language model p_i and a set of texts $\mathcal{D}_N = \{x_1, \dots, x_N\}$ sampled from a population $\mathcal{D}^\dagger$, we define the log-likelihood vector as

$$\ell_i = (\log p_i(x_1), \dots, \log p_i(x_N))^\top \in \mathbb{R}^N.$$

Let the vector of centered log-likelihoods, with the effect of length removed¹, be denoted by $\xi_i \in \mathbb{R}^N$. Oyama et al. (2025b) showed that the squared Euclidean distance between ξ_i and ξ_j approximates the Kullback–Leibler (KL) divergence between p_i and p_j . A language model map (Figure 1) is obtained by applying dimensionality reduction to these vectors.

Inefficiency of uniform sampling. Typically, $\mathcal{D}_N$ is constructed via uniform sampling. Oyama et al. (2025a) showed that the informativeness of texts is not uniform: texts with higher inter-model log-likelihood variance

$$V(x; \mathcal{M}) = \text{Var}_{p_i \in \mathcal{M}}(\xi_i(x))$$

are more model-discriminative, and resampling with probability proportional to $V(x; \mathcal{M})$ achieves the same KL estimation accuracy using roughly half the number of texts. This means that approximately half of the texts selected by uniform sampling are effectively redundant.

Limitations of prior work. However, this finding identifies the inefficiencies of uniform sampling, but does not address the question of which texts should be selected. We focus on the remapping setting, in which a map is constructed on a

¹ $\xi_i(x) = \log p_i(x) - \bar{m}_i L(x)$, where $\bar{m}_i = \sum_x \log p_i(x) / \sum_x L(x)$ is the mean log-likelihood per unit length and $L(x)$ is the text length. Since log-likelihoods scale roughly with text length, we remove this component.

new domain, such as medicine or law. If we could identify informative texts in advance and use them to construct the text set $\mathcal{D}_N$, we could reduce the computational cost while improving the accuracy of the estimation. To achieve this, a method to predict $V(x; \mathcal{M})$ is required before the computation of log-likelihoods for target LLMs.

3 Length and Meaning Explain Variance

In this section, as a preliminary analysis for variance prediction, we examine how well the variance is explained by two aspects of texts: their *length* and *meaning* (captured by embeddings).

Observation 1: Length explains variance, but not fully. Figure 3 (Left) shows a scatter plot of text length $L(x)$ and variance $V(x; \mathcal{M})$ ². Points are colored by the residual $\Delta z(x) = \log V(x; \mathcal{M}) - \log L(x)$. Although variance tends to increase in proportion to text length, there is a wide range of variation within the same length, indicating that length does not fully explain variance.

Observation 2: Meaning captures the residual. Figure 3 (Center) shows a t-SNE visualization of the text embeddings. Texts with large residuals and small residuals form clusters in the embedding space, indicating that the variance structure not captured by length is reflected in the embeddings.

Our approach. Based on these observations, text length and embeddings provide complementary information regarding likelihood variance. Therefore, we propose a predictor that incorporates both to predict $V(x; \mathcal{M})$. We introduce PredIS, which uses the predicted variance to select texts in Section 4.

²Variance is computed over $|\mathcal{M}| = 99$ LLMs (described in Appendix A) on $|\mathcal{D}^*| = 10,000$ texts from the Pile. Text length $L(x)$ is consistently measured by the GPT-2 tokenizer.

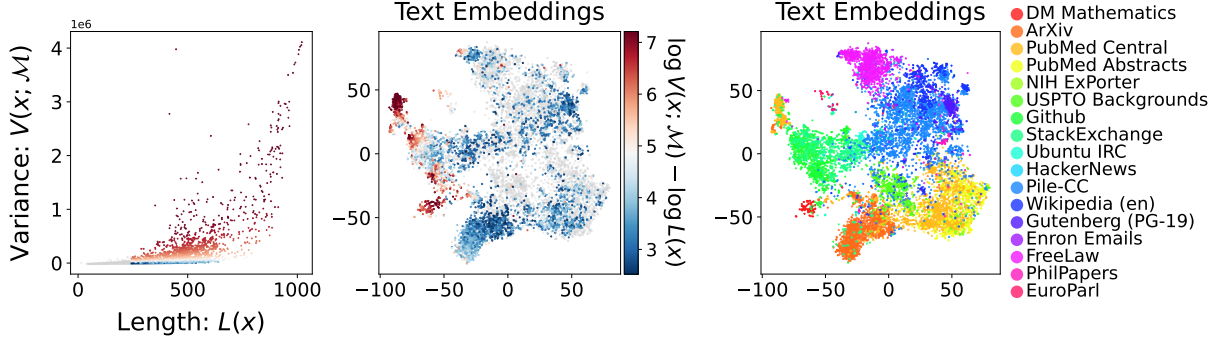


Figure 3: **Likelihood variance $V(x; \mathcal{M})$ is explained from text length and meaning (embeddings).** **Left Panel:** Scatter plot of variance $V(x; \mathcal{M})$ and length $L(x)$, colored by $\Delta z(x) = \log V(x; \mathcal{M}) - \log L(x)$. Short texts ($\log L(x) < 5.5$) are shown in gray. **Center Panel:** t-SNE of text embeddings colored by $\Delta z(x)$. **Right Panel:** Same t-SNE colored by text category in the Pile. High- Δz (red) and low- Δz (blue) texts form distinct clusters corresponding to specific categories, showing that text embeddings reflect meaning-level variance beyond length.

4 Methods

This section describes the components of PredIS in order: training the variance predictor (Section 4.1), text sampling and KL divergence estimation using the predicted variance (Section 4.2), and the evaluation protocol for sampling methods (Section 4.3).

4.1 Importance Predictor

Training. We train a linear regression model³ f that predicts the log variance $\log V(x; \mathcal{M})$ from a 769-dimensional input vector formed by concatenating $\log L(x)$ and a text embedding $\mathbf{v}(x) \in \mathbb{R}^{768}$. The embedding is obtained from SimCSE-RoBERTa-base (Gao et al., 2021). We predict on a log scale because $V(x; \mathcal{M})$ varies over several orders of magnitude. We split $\mathcal{D}^*$ into training $\mathcal{D}_{\text{train}}^*$ and validation $\mathcal{D}_{\text{val}}^*$ sets in an 8:2 ratio, and train the predictor on log-likelihood data from 99 LLMs. Since linear regression is sensitive to outliers, we examine the norms $\|\xi_i\|$ of the centered log-likelihood vectors on $\mathcal{D}_{\text{train}}^*$, and exclude DialoGPT-small as an outlier. This process is performed at predictor-training time with no additional cost. Ablations with and without this exclusion are reported in Appendix B.

Validation. We denote the predicted variance for a text x as $\hat{V}(x) = \exp(f(\mathbf{v}(x), \log L(x)))$. Figure 4 shows a scatter plot of the ground-truth variance $V(x; \mathcal{M})$ and the predicted variance $\hat{V}(x)$ on $\mathcal{D}_{\text{val}}^*$. The high Spearman correlation ($\rho = 0.862$)

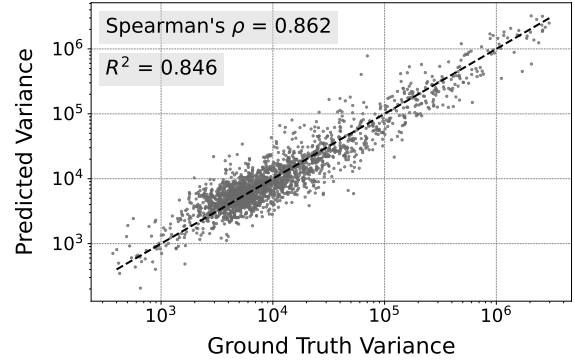


Figure 4: **Validation of the variance predictor.** Scatter plot of ground-truth variance $V(x; \mathcal{M})$ and predicted variance $\hat{V}(x)$ on the 2,000 test texts. Both axes are in log scale. The high correlation (Spearman $\rho = 0.862$) demonstrates that the predictor effectively estimates the likelihood variance across models.

indicates that the variance can be effectively predicted from the text length and embedding.

4.2 Sampling and Estimation

Sampling. Based on the predicted variance $\hat{V}(x)$, we define the sampling probability of a text $x \in \mathcal{D}^\dagger$ as⁴

$$\pi^{\text{PredIS}}(x) = \underbrace{\frac{(1 - \alpha) \hat{V}(x)^\tau}{\sum_{x' \in \mathcal{D}^\dagger} \hat{V}(x')^\tau}}_{\text{Predicted Variance}} + \underbrace{\frac{\alpha}{|\mathcal{D}^\dagger|}}_{\text{Uniform}}.$$

The first term is proportional to $\hat{V}(x)$ smoothed by an exponent $\tau \in (0, 1]$, and the second term

³We adopt a linear model because it is the simplest choice and, as shown below, achieves sufficient accuracy.

⁴Since $\hat{V}(x)$ has a wide dynamic range and the sampling probabilities tend to take extreme values, we introduce smoothing and uniform mixture. See Appendix C for details.

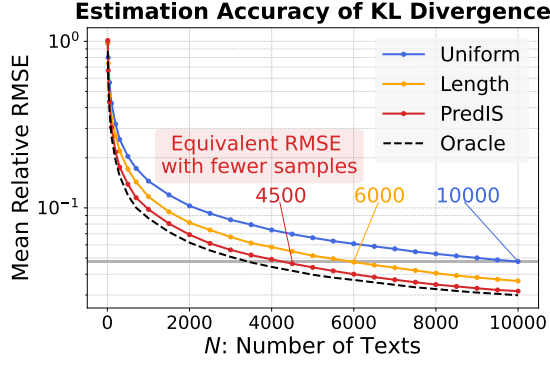


Figure 5: **Mean Relative RMSE and sample size N on the Pile (lower is better).** The horizontal dashed line marks Uniform at $N = 10,000$. Annotations indicate the N values at which Length ($N = 6,000$) and PredIS ($N = 4,500$) match this accuracy. Oracle (dashed) serves as an approximate upper bound.

is a uniform mixture corresponding to defensive importance sampling. We sample N texts from π^{PredIS} without replacement to construct $\mathcal{D}_N^{\text{PredIS}}$.

Estimation of KL Divergence. For each sampled text $x \in \mathcal{D}_N^{\text{PredIS}}$, we compute the log-likelihood $\log p_i(x)$ of each language model p_i . We then apply weighted centering⁵ to obtain

$$\hat{\xi}_i(x) = \log p_i(x) - \hat{m}_i L(x),$$

and estimate the population-level distance $g_{ij} = \sum_{x \in \mathcal{D}^\dagger} (\xi_i(x) - \xi_j(x))^2$ by

$$\hat{g}_{ij} = \sum_{x \in \mathcal{D}_N^{\text{PredIS}}} \tilde{w}(x) (\hat{\xi}_i(x) - \hat{\xi}_j(x))^2.$$

Here, $\tilde{w}(x) = w(x) / \sum_{x' \in \mathcal{D}_N^{\text{PredIS}}} w(x')$ is the normalized importance weight and $w(x) = 1/\pi(x)$. This estimator follows the Self-Normalized Importance Sampling (SNIS) framework (Owen, 2013)⁶.

4.3 Evaluation of Sampling Methods

4.3.1 Baselines

We compare the performance of PredIS with the following three baselines.

(1) Uniform: samples all texts with uniform probability,

$$\pi^{\text{Unif}}(x) = \frac{1}{|\mathcal{D}^\dagger|}.$$

⁵The mean log-likelihood per unit length is estimated as $\hat{m}_i = \sum_{x \in \mathcal{D}_N^{\text{PredIS}}} w(x) \log p_i(x) / \sum_{x \in \mathcal{D}_N^{\text{PredIS}}} w(x) L(x)$.

⁶We assume approximate independence under sampling without replacement, which is a commonly used approximation when the population size $|\mathcal{D}^\dagger|$ is sufficiently large.

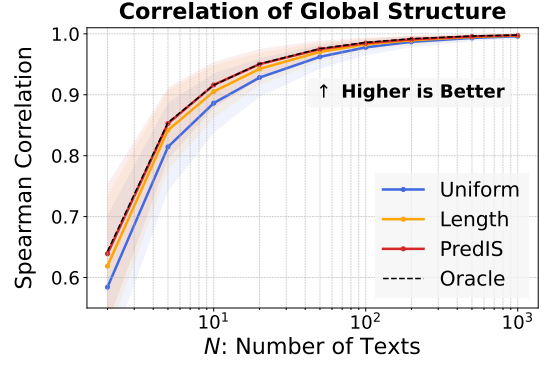


Figure 6: **Spearman correlation between the estimated pairwise KL matrix $\hat{G}^{(t)}$ and oracle G on the Pile, and sample size N .** Bands indicate $\bar{\rho} \pm \text{std}$ across trials. PredIS achieves higher correlation than Uniform and Length at all N .

(2) Length: samples each text with probability proportional to its length $L(x)$,

$$\pi^{\text{Length}}(x) \propto L(x).$$

(3) Oracle: samples with probability proportional to the true variance $V(x; \mathcal{M})$,

$$\pi^{\text{Oracle}}(x) = \frac{(1 - \alpha) V(x; \mathcal{M})^\tau}{\sum_{x' \in \mathcal{D}^\dagger} V(x'; \mathcal{M})^\tau} + \frac{\alpha}{|\mathcal{D}^\dagger|}.$$

This corresponds to the case where the variance is perfectly predicted, and serves as an upper bound on performance. We use the same values of τ and α as in PredIS.

For each Method (Unif, Length, PredIS, and Oracle), we sample $\mathcal{D}_N^{\text{Method}}$ from the corresponding π^{Method} and compute $\hat{g}_{ij}$.

4.3.2 Evaluation Metrics

Estimation Accuracy. To evaluate the estimation accuracy of g_{ij} , the population-level KL divergence between model p_i and p_j , we perform T independent trials and compute the Mean Squared Error (MSE) of the estimator $\hat{g}_{ij}$:

$$\text{MSE}_{ij} = \frac{1}{T} \sum_{t=1}^T (\hat{g}_{ij}^{(t)} - g_{ij})^2.$$

To compare the accuracy of KL estimations across model pairs in a scale-invariant manner, we use the relative RMSE, defined as

$$e_{ij} := \frac{\sqrt{\text{MSE}_{ij}}}{g_{ij}}. \quad (1)$$

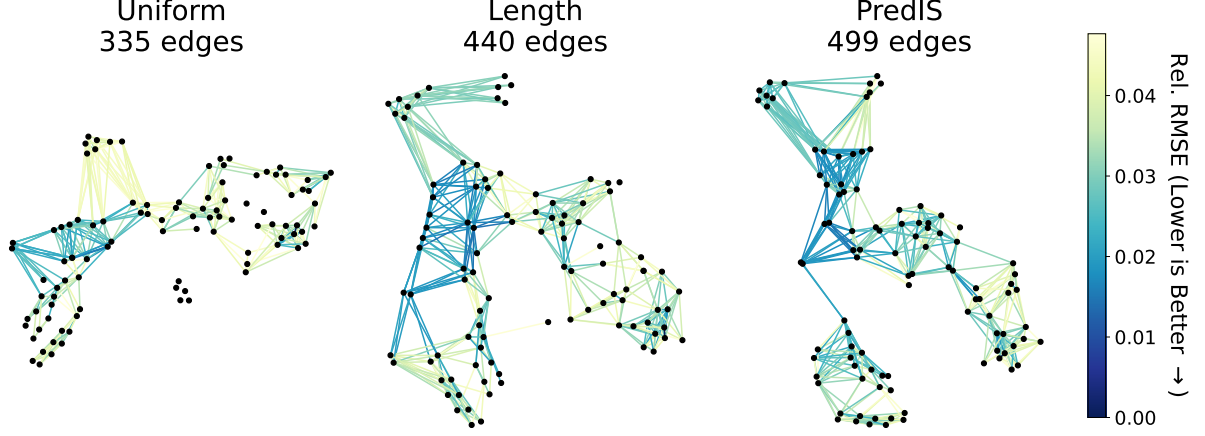


Figure 7: **Language model maps constructed by each sampling method** ($N = 10,000$, the Pile). Each point is one of 100 models, positioned by t-SNE on the corresponding log-likelihood vector. An edge (i, j) is drawn when the pair is among the $k = 10$ nearest neighbors and its Relative RMSE $\leq \gamma = 0.0477$ (γ is the mean Relative RMSE of Uniform). Edge color indicates Relative RMSE. PredIS (499 edges) preserves the most accurately estimated neighbor pairs, followed by Length (440) and Uniform (335).

Efficiency Gain. To evaluate the improvement in MSE compared to a baseline, we use the Efficiency Gain defined as

$$R_{ij} = \frac{\text{MSE}_{ij}^{\text{Uniform}}}{\text{MSE}_{ij}^{\text{Method}}}. \quad (2)$$

Here, $R_{ij} > 1$ indicates that the sampling method achieves a smaller MSE than the baseline for the KL estimation of the corresponding model pair. The Efficiency Gain R_{ij} also implies that the baseline would require approximately $R_{ij} \times N$ samples to achieve the same estimation accuracy as the proposed method, when both use N samples.

Global Structure. To evaluate whether KL estimation accuracy contributes to preserving the relative positions among models, we use the Spearman rank correlation between the upper-triangular entries ($\binom{K}{2}$ pairs) of the estimated distance matrix $\hat{\mathbf{G}}^{(t)} \in \mathbb{R}^{K \times K}$ and the oracle $\mathbf{G} \in \mathbb{R}^{K \times K}$, averaged over T trials:

$$\bar{\rho} = \frac{1}{T} \sum_{t=1}^T \text{Spearman}(\hat{\mathbf{G}}^{(t)}, \mathbf{G}). \quad (3)$$

We use the Spearman correlation coefficient instead of the Pearson correlation coefficient because we focus on the relative ordering of the pairwise distances rather than their absolute values. This is consistent with the framework of Representational Similarity Analysis (RSA) (Kriegeskorte et al., 2008).

5 Experiments

Section 5.1 verifies the sampling efficiency improvement from variance prediction on $\mathcal{D}_{\text{pile}}^{\dagger}$, and Section 5.2 shows that the predictor generalizes to the other 24 domains without retraining.

Setup. We evaluate the sampling methods on $K = 100$ LLMs from the Hugging Face Hub. For each of 25 domains, we construct a population $\mathcal{D}_{\text{domain}}^{\dagger}$ of 10^6 texts, from which each sampling method draws texts⁷. The 25 domains consist of the Pile and 24 domains corresponding to the textual-domain categorization of Common Pile v0.1 (Kandpal et al., 2025). The predictor trained in Section 4.1 is based on $\mathcal{D}^*$, and all populations $\mathcal{D}_{\text{domain}}^{\dagger}$ used for evaluation are disjoint from it. We report the mean and standard deviation of the evaluation metrics across $T = 1,000$ independent trials.

5.1 Efficiency via Variance Prediction

KL divergence estimation accuracy. Figure 5 shows the relationship between mean relative RMSE (Eq. 1) and the sample size N . PredIS consistently achieves a lower relative RMSE than Uniform and Length. To match the accuracy of Uniform at $N = 10,000$, Length requires $N = 6,000$ texts and PredIS requires $N = 4,500$ texts.

Distance estimation for neighboring pairs on the model map. For each sampling method, we sample $N = 10,000$ texts and construct a t-SNE map

⁷See Appendix A for the full list of 100 models and construction details of population of 25 domains.

Domain of $\mathcal{D}_{\text{domain}}^{\dagger}$	Mean Rel. RMSE ($\downarrow$)				Median Efficiency Gain ($\uparrow$)			Spearman Corr. ($\uparrow$)			
	Unif	Length	PredIS	Oracle	Length	PredIS	Oracle	Unif	Length	PredIS	Oracle
the Pile	0.048	0.036	0.032	0.030	1.72	2.04	2.16	0.886 ± 0.049	0.905 ± 0.043	0.916 ± 0.038	0.916 ± 0.039
arxiv abstracts	0.025	0.022	0.022	0.019	1.22	1.24	1.57	0.894 ± 0.029	0.903 ± 0.029	0.903 ± 0.029	0.908 ± 0.026
arxiv papers	0.046	0.040	0.029	0.027	1.15	1.81	1.93	0.903 ± 0.037	0.907 ± 0.036	0.920 ± 0.030	0.921 ± 0.032
biodiversity heritage library	0.028	0.024	0.022	0.019	1.20	1.30	1.69	0.872 ± 0.034	0.877 ± 0.034	0.880 ± 0.035	0.889 ± 0.032
caselaw access project	0.027	0.024	0.022	0.020	1.26	1.37	1.58	0.900 ± 0.034	0.908 ± 0.032	0.909 ± 0.031	0.913 ± 0.029
cccc	0.037	0.031	0.028	0.025	1.32	1.52	1.97	0.853 ± 0.045	0.862 ± 0.042	0.870 ± 0.041	0.870 ± 0.039
data provenance initiative	0.039	0.030	0.029	0.026	1.30	1.25	1.61	0.850 ± 0.049	0.869 ± 0.047	0.871 ± 0.047	0.884 ± 0.039
doab	0.053	0.036	0.031	0.026	1.84	2.32	2.72	0.890 ± 0.034	0.900 ± 0.033	0.904 ± 0.035	0.904 ± 0.034
github archive	0.156	0.217	0.108	0.042	1.40	1.72	3.36	0.872 ± 0.053	0.892 ± 0.046	0.899 ± 0.040	0.898 ± 0.041
library of congress	0.042	0.038	0.034	0.026	1.18	1.31	2.65	0.856 ± 0.047	0.862 ± 0.044	0.867 ± 0.043	0.877 ± 0.041
peS2o	0.037	0.030	0.026	0.023	1.22	1.34	1.73	0.909 ± 0.028	0.913 ± 0.026	0.912 ± 0.026	0.915 ± 0.027
pre 1929 books	0.044	0.037	0.033	0.026	1.22	1.40	2.46	0.859 ± 0.048	0.865 ± 0.045	0.869 ± 0.047	0.886 ± 0.042
project gutenberg	0.047	0.037	0.032	0.029	1.52	1.80	2.59	0.888 ± 0.042	0.900 ± 0.043	0.921 ± 0.036	0.924 ± 0.033
pubmed	0.023	0.020	0.023	0.021	1.51	1.10	1.37	0.927 ± 0.053	0.957 ± 0.031	0.955 ± 0.019	0.958 ± 0.021
regulations	0.060	0.043	0.040	0.035	1.72	1.40	1.59	0.915 ± 0.053	0.936 ± 0.039	0.932 ± 0.036	0.933 ± 0.041
stackexchange	0.114	0.140	0.072	0.037	1.42	2.41	3.60	0.918 ± 0.035	0.931 ± 0.031	0.938 ± 0.028	0.937 ± 0.026
stackv2 edu	0.056	0.054	0.052	0.039	1.17	1.14	1.70	0.924 ± 0.031	0.925 ± 0.030	0.923 ± 0.028	0.926 ± 0.025
stackv2 html	0.044	0.037	0.033	0.028	1.35	1.46	2.10	0.872 ± 0.050	0.884 ± 0.045	0.887 ± 0.045	0.887 ± 0.041
ubuntu irc	0.061	0.055	0.045	0.026	1.18	1.61	4.28	0.854 ± 0.037	0.858 ± 0.035	0.858 ± 0.036	0.862 ± 0.036
uk hansard	0.054	0.040	0.030	0.029	1.86	2.60	3.15	0.800 ± 0.050	0.809 ± 0.051	0.844 ± 0.061	0.870 ± 0.059
usgpo	0.155	0.101	0.074	0.048	1.92	2.18	2.91	0.907 ± 0.049	0.922 ± 0.035	0.927 ± 0.024	0.930 ± 0.026
uspto	0.061	0.042	0.041	0.040	2.20	2.47	1.90	0.863 ± 0.047	0.876 ± 0.051	0.898 ± 0.043	0.900 ± 0.043
wikimedia	0.061	0.049	0.040	0.029	1.24	1.47	2.40	0.888 ± 0.038	0.899 ± 0.036	0.900 ± 0.036	0.901 ± 0.037
wikiteam	0.063	0.052	0.042	0.029	1.41	1.70	2.59	0.892 ± 0.042	0.905 ± 0.037	0.906 ± 0.038	0.908 ± 0.033
youtube	0.232	0.170	0.099	0.045	1.89	5.54	16.72	0.858 ± 0.041	0.858 ± 0.038	0.857 ± 0.039	0.860 ± 0.055

Table 1: **Comparison of sampling methods across 25 text domains.** The 24 domains other than the Pile correspond to the textual-domain categorization of Common Pile v0.1. Mean relative RMSE and median Efficiency Gain are computed at $N = 10,000$. Spearman ρ between the estimated and oracle pairwise KL distance matrices is reported at $N = 10$ (mean $\pm$ std). The predictor is trained on $\mathcal{D}_{\text{train}}^*$ and applied to all other domains without retraining. Oracle serves as an approximate upper bound. **Bold** indicates the best non-Oracle result per metric.

of the 100 LLMs. In Figure 7, we connect $k = 10$ nearest-neighbor pairs on each map if their relative RMSE is below the average relative RMSE of Uniform on the Pile, $\gamma = 0.0477$. The number of such edges is 335 for Uniform, 440 for Length, and 499 for PredIS, indicating that PredIS estimates distances between neighboring pairs most accurately.

Preservation of global structure. Figure 6 shows the relationship between Spearman correlation $\bar{\rho}$ (Eq. 3) and the sample size N . PredIS outperforms Uniform and Length across all N , best preserving the relative ordering of pairwise model distances.

5.2 Domain Generalization

We apply the predictor, trained on $\mathcal{D}_{\text{train}}^*$, to 24 other domains without retraining to verify the generalization of PredIS. Table 1 summarizes the results across all 25 domains.

Accuracy. PredIS outperforms both Uniform and Length in mean relative RMSE on 23 of the 24 domains. The only exception is pubmed, where the RMSE of PredIS (0.023) matches Uniform and is worse than Length (0.020). However, Oracle on pubmed (0.021) also only performs on an equal level with Length, suggesting that the variance distribution on this domain differs from the others.

Efficiency Gain. PredIS achieves a higher median Efficiency Gain than Length on 20 of the 24 domains. Notable improvements appear on youtube (5.54), uk_hansard (2.60), and stackexchange (2.41). This suggests that the patterns of text importance for the model set are shared across domains, and that a predictor based on text embeddings can capture these patterns broadly.

Global Structure. PredIS also achieves the highest Spearman correlation ($N = 10$) on the majority of domains. On uk_hansard, the gap from Uniform reaches 0.044, showing that PredIS effectively preserves the relative ordering of model distances even at small sample sizes. Appendix D reports Spearman correlations for $N \in \{100, 1,000, 10,000\}$.

Summary. PredIS consistently outperforms the baselines across most of the 24 domains, with degradation observed on only one domain. This demonstrates that a single trained variance predictor functions robustly across diverse domains. The predictor used here is trained without the outlier model. In Appendix B, we compare it with a predictor trained with the outlier included. Overall, both performs similarly, but the excluding the outlier results in slightly more stable performance.

Model	Uniform	Length	PredIS
	$N=10,000$	$N=6,000$	$N=4,500$
Qwen3-0.6B	4.6 min	2.8 min	<u>2.2 min</u>
Qwen3-1.7B	4.7 min	2.8 min	<u>2.2 min</u>
Qwen3-4B	6.0 min	3.7 min	<u>2.9 min</u>
Qwen3-8B	6.6 min	4.1 min	<u>3.3 min</u>
Qwen3-14B	9.9 min	6.2 min	<u>4.9 min</u>

(a) Log-likelihood computation time.

Model	vs. Uniform		vs. Length	
	b	K^*	b	K^*
Qwen3-0.6B	+2.4 min	13	+0.6 min	50
Qwen3-1.7B	+2.5 min	12	+0.6 min	50
Qwen3-4B	+3.1 min	10	+0.8 min	38
Qwen3-8B	+3.3 min	10	+0.8 min	38
Qwen3-14B	+5.0 min	6	+1.3 min	24

(b) Break-even analysis.

Table 2: **Computational cost of PredIS.** (a) Log-likelihood computation time on a single NVIDIA A100 (40GB) GPU. Length ($N = 6,000$) and PredIS ($N = 4,500$) use smaller subsets, chosen to match the Mean Relative RMSE of Uniform at $N = 10,000$ (Figure 5). The fastest method per row is underlined. (b) Break-even analysis of the offline cost $a = 30$ min, measured as the embedding computation time for the $|\mathcal{D}^\dagger| = 10^6$ texts of the Pile population. b is the per-model time saving of PredIS in the accuracy-matched settings of (a), and $K^* = \lceil a/b \rceil$ is the break-even number of models. It is the minimum number of models required to offset the offline cost.

6 Discussion

In this section, we first discuss the computational efficiency of PredIS (Section 6.1). We then discuss why PredIS works from the optimality of the sampling distribution (Section 6.2) and the content of the selected texts (Section 6.3).

6.1 Computational Efficiency of PredIS

Log-likelihood computation cost. Table 2a shows the log-likelihood computation time for each method (Qwen3 at various sizes, single NVIDIA A100 40GB). At settings that match the estimation accuracy of Uniform ($N = 10,000$), PredIS consistently achieves the shortest computation, with about a 50% reduction on Qwen3-14B (from 9.9 to 4.9 min). This reduction is consistent with the median Efficiency Gain (Eq. 2) over Uniform on the Pile, which is 2.04 for PredIS (Table 1).

Offline cost and break-even analysis. PredIS requires the computation of embeddings for all texts in population $\mathcal{D}^\dagger$ prior to sampling. In contrast, Uniform and Length do not require such processing. We therefore compare the methods in terms of the total computational cost including this offline cost. In our setting, computing the embeddings of $|\mathcal{D}^\dagger| = 10^6$ texts took about $a = 30$ minutes on a single A100 40GB GPU. In these accuracy-matched settings, letting t denote the per-model log-likelihood computation time, the total cost of evaluating K models is $a + K t_{\text{PredIS}}$ for PredIS and $K t_{\text{Baseline}}$ for a baseline. Letting $b = t_{\text{Baseline}} - t_{\text{PredIS}}$ be the per-model time saving of PredIS over the baseline, we have

$$K t_{\text{Baseline}} \geq a + K t_{\text{PredIS}} \iff K \geq a/b.$$

Therefore, the offline cost is offset once at least $K^* = \lceil a/b \rceil$ models are evaluated. We call K^* the break-even number of models. Based on the measurements in Table 2a, we compute K^* against Uniform and against Length.

Comparison with Uniform. The “vs. Uniform” column in Table 2b shows the per-model time savings (b) and the number of models (K^*) needed to break even against Uniform ($N = 10,000$). Depending on model size, K^* ranges from 6 to 13. In other words, even for models comparable to Qwen3-0.6B, evaluating just 13 models is sufficient to offset the offline cost of the embedding computation. Smaller models yield smaller per-model savings and thus require more evaluations to break even. In contrast, larger models offset the cost with fewer evaluations.

Comparison with Length. The “vs. Length” column in Table 2b shows b and K^* against Length ($N = 6,000$). Since Length is itself more efficient than Uniform, the savings are smaller, and K^* ranges from 24 to 50. For models comparable to Qwen3-0.6B or 1.7B, PredIS becomes advantageous over Length in terms of the total computational cost when evaluating more than 50 models.

Summary. From a break-even perspective, PredIS becomes advantageous when a sufficient number of models are evaluated. In this analysis, the break-even number K^* ranges from 6 to 50, which is less than the $K = 100$ models evaluated. However, K^* depends on the computational setup, such as the embedding model and the population size, and can be larger in other settings. This break-even perspective also provides a practical guideline for practitioners when choosing a sampling method.

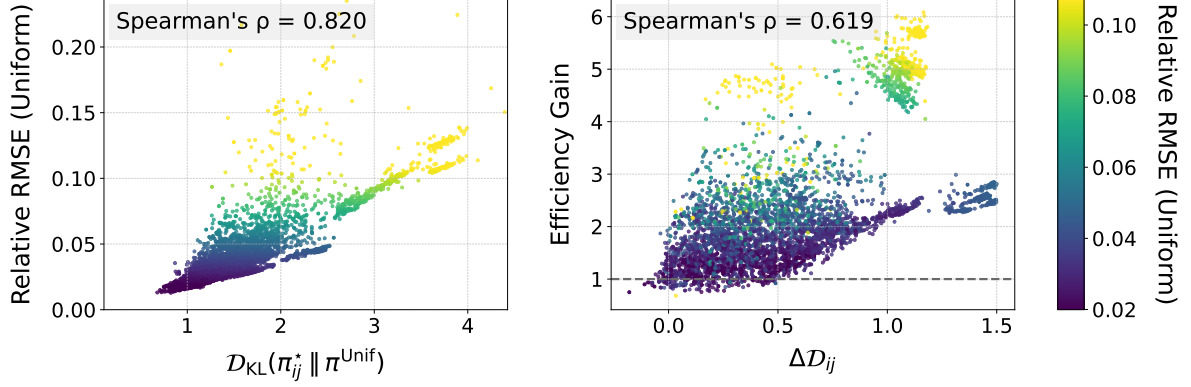


Figure 8: **Analysis of PredIS via sampling optimality.** Each point represents a model pair (p_i, p_j) (100 models, 4,947 valid pairs). **Left Panel:** Scatter plot between KL divergence of the optimal distribution from uniform $D_{\text{KL}}(\pi_{ij}^* \parallel \pi^{\text{Unif}})$ and Relative RMSE of Uniform. Pairs harder to estimate under uniform sampling deviate more from uniform in their optimal distribution. **Right Panel:** Scatter plot between Divergence Reduction ΔD_{ij} and Efficiency Gain R_{ij} . Approaching the optimal distribution correlates with higher estimation efficiency (Spearman $\rho = 0.619$) and only 3.3% of pairs fall below the dashed line ($R_{ij} < 1$).

6.2 Analysis via Divergence Reduction

Optimal distribution. The theoretically optimal sampling distribution π_{ij}^* that minimizes the variance of the estimator g_{ij} for a given model pair (p_i, p_j) is known to be proportional to the squared discrepancy term:

$$\pi_{ij}^*(x) \propto (\xi_i(x) - \xi_j(x))^2.$$

While π_{ij}^* is unique to each pair and cannot be matched by a single predictor for all pairs, approaching this optimal distribution is key to variance reduction (Bugallo et al., 2017). Specifically, Jerfel et al. (2021) argue that minimizing the forward KL divergence $D_{\text{KL}}(\pi^* \parallel \pi)$ is appropriate from the perspective of variance suppression in importance sampling.

Divergence reduction. To quantify how much closer π^{PredIS} moves to π_{ij}^* compared to π^{Unif} , we define

$$\Delta D_{ij} := D_{\text{KL}}(\pi_{ij}^* \parallel \pi^{\text{Unif}}) - D_{\text{KL}}(\pi_{ij}^* \parallel \pi^{\text{PredIS}}),$$

where the KL divergence on the population $\mathcal{D}^\dagger$ is defined as $D_{\text{KL}}(\pi_{ij}^* \parallel \pi) = \sum_{x \in \mathcal{D}^\dagger} \pi_{ij}^*(x) \log(\pi_{ij}^*(x)/\pi(x))$. $\Delta D_{ij} > 0$ means that PredIS is closer to the optimal distribution than Uniform. We analyze the relationship between ΔD_{ij} and estimation accuracy on $\mathcal{D}_{\text{pile}}^\dagger$ for 4,947 model pairs⁸.

⁸Three pairs out of $\binom{100}{2} = 4,950$ are excluded because $\xi_i(x) = \xi_j(x)$ for all x , making π_{ij}^* undefined.

Harder pairs deviate more from uniform. The harder it is to estimate a model pair, the more its optimal distribution differs from the uniform distribution. Figure 8 (Left) shows the relationship between the divergence of the optimal distribution from uniform, $D_{\text{KL}}(\pi_{ij}^* \parallel \pi^{\text{Unif}})$, and the relative RMSE of Uniform sampling. A positive correlation is observed (Spearman $\rho = 0.820$), indicating that the difficulty of estimating a pair is characterized by how far its pair-specific optimal distribution deviates from uniform.

Closer to optimal results in better accuracy. As PredIS approaches the optimal distribution, the estimation error is reduced. Figure 8 (Right) shows the relationship between the Divergence Reduction ΔD_{ij} and the Efficiency Gain R_{ij} . A positive correlation is observed (Spearman $\rho = 0.619$), indicating that approaching the optimal distribution leads to improved estimation. For a small fraction of pairs ($\approx 3.3\%$) where ΔD_{ij} is low or negative, performance degrades ($R_{ij} < 1$), but PredIS is effective for most pairs.

6.3 Length and Meaning

In this section, we qualitatively observe which texts in the Pile population are prioritized or excluded by PredIS for the model set $\mathcal{M}$ used in this work. In addition to $\Delta z(x)$, we use the mean centered log-likelihood $\bar{\xi}(x) := \sum_{p_i \in \mathcal{M}} \xi_i(x)/|\mathcal{M}|$, which measures how easily models in $\mathcal{M}$ assign high probability to a text on average. Table 3 shows representative texts from the top 10% and bottom 10% of $\Delta z(x)$, further divided by $\bar{\xi}(x)$. Texts with high

Condition	Category	Example Text
High $\Delta z(x)$	Pile-CC (21.1%)	How Magnets Work Whether you're sticking them to the fridge or using them to transpose sound to tap... 57 tokens
	Github (17.0%)	o, Type> InitializeGetStateMachineType() { Type asyncStateMachineAttribute = Type.GetType("Syst... 301 tokens
	Wikipedia (en) (12.7%)	Cymodema Cymodema is a genus of true bugs in the family Cymidae. There are about seven described sp... 166 tokens
Low $\Delta z(x)$ High $\bar{\xi}(x)$	ArXiv (67.2%)	putationdata/nonlinear\{}_problem/Laplaceto2.db "SELECT DISTINCT DOFS\{}_primal, relativeError3 from da... 454 tokens
	StackExchange (13.8%)	Q: Run php script without cron to do automatic database backup I am trying to run a php script wit... 367 tokens
	Github (9.0%)	echo_red "Unable to download headless VGA BIOS; headless firmware cannot be installed." else ... 500 tokens
Low $\Delta z(x)$ Low $\bar{\xi}(x)$	ArXiv (39.5%)	0H\$. The power spectrum of tensor perturbations is normalized such that in a critical density univer... 308 tokens
	PubMed Central (17.5%)	econdary structure ([Fig. 3b](#F3){ref-type="fig"}). This provides little evidence for conservation ... 242 tokens
	Gutenberg (PG-19) (12.6%)	r, both of them silent; till Daisy put her lips to the fine olive-dark cheek of the old woman and ki... 279 tokens

Table 3: **Representative texts for each group defined by $\Delta z(x)$ and $\bar{\xi}(x)$.** High- Δz texts (top 10%) have high inter-model variance and are prioritized by PredIS. They include Pile-CC web articles, GitHub code, and Wikipedia entries. Low- Δz texts (bottom 10%) are further divided by $\bar{\xi}$: those with high $\bar{\xi}$ (easy for all models) are dominated by structured content such as ArXiv code, database queries, and shell scripts (67% ArXiv); those with low $\bar{\xi}$ (hard for all models) consist of highly specialized or archaic text such as ArXiv equations, PubMed papers, and Project Gutenberg literature. Category percentages are computed within each group. Texts are truncated to 100 characters.

$\Delta z(x)$ are highly discriminative between models and are prioritized by PredIS. In contrast, texts with low $\Delta z(x)$ fall into two types: easy-to-predict formulaic content that all models handle well, and specialized or archaic content that all models struggle with. Both types are weakly discriminative. Crucially, these low- Δz texts are often indistinguishable from each other by length alone. PredIS captures this content structure in the embedding space and samples in a way that reflects variance differences not explained by length.

7 Related Work

Comparing many language models. Over 300,000 language models are publicly available on the Hugging Face Hub, and various studies have proposed ways to quantify and systematically compare them (Zhu et al., 2025; Yax et al., 2025; Zhuang et al., 2025; Horwitz et al., 2025; Zhang and Bollegala, 2026). We focus on reducing the computational cost of language model maps (Oyama et al., 2025b).

Efficient Model evaluation. Active testing (Kossen et al., 2021, 2022; Berrada et al., 2025) selects test points non-uniformly with a surrogate model and importance-weights the estimator, while benchmark subset selection (Maia Polo et al.,

2024; Vivek et al., 2024) samples representative examples from the responses of the target models. PredIS shares the structure of active testing, but targets pairwise KL divergences among many models rather than the performance of a single model, and selects texts before running the target models.

Selecting informative data. The idea of prioritizing samples on which models disagree shares motivation with Query-by-Committee (Seung et al., 1992) and the discrimination parameter in Item Response Theory (Baker, 2001). We apply this principle to the comparison of LLMs.

8 Conclusion

We addressed the computational cost of language model maps by proposing Predictive Importance Sampling (PredIS). Experiments on 100 LLMs across 25 domains showed that PredIS consistently outperforms baselines and that a predictor trained on the Pile generalizes to a wide range of domains without retraining. Our analysis further revealed that text importance cannot be explained by text length alone; it is closely related to meaning reflected in the embedding space. This work provides a foundation for efficiently comparing the rapidly growing number of language models.

Limitations

- **Offline cost of PredIS.** PredIS requires the computation of embeddings for all texts in population. The cost is about 30 minutes for $|\mathcal{D}^\dagger| = 10^6$ texts in our setting. While the break-even number of models K^* ranges from 6 to 50 in Section 6.1, it can be larger depending on the domain, the embedding model, and the hardware. When only a small number of models are evaluated, Uniform or Length can be preferable in terms of the total computational cost.
- **Generalization across model sets.** This study focuses on the remapping setting, where the target model set $\mathcal{M}$ is fixed. The optimality analysis in Section 6.2 suggests that PredIS is effective for most pairs in $\mathcal{M}$, indicating a degree of robustness against changes in the target models. However, a systematic study of how predictor generalization extends across different model sets is beyond the scope of this work. Future directions include constructing a core set common to broader model collections and analyzing the characteristics of model-specific discriminative texts.
- **Language limitation.** Our verification is limited to English text domains. While we tested across diverse domains (code, law, medicine), the applicability of our embedding-based predictor to multilingual or non-English settings remains unverified.
- **Choice of embeddings.** We employed a single embedding model (SimCSE) to predict importance. Exploring other embedding architectures or fine-tuning them specifically for variance prediction could potentially improve accuracy, which we leave for future work.
- **Hyperparameter optimization in importance sampling.** In our implementation of importance sampling, we fixed the smoothing exponent τ and the uniform mixture coefficient α at empirical values ($\tau = 0.5$, $\alpha = 0.1$) without extensive optimization (Appendix C). Since our primary focus is on leveraging predicted variances, these adjustments are sufficient for effective text selection. Adaptively tuning τ and α according to the characteristics of the proposal distribution could yield further performance improvements.

- **Closed-source models.** Our framework requires access to per-text log-likelihoods, which are typically not exposed by closed-source LLMs. Our analysis is therefore restricted to open-source LLMs available on the Hugging Face Hub.

Acknowledgments

This study was partially supported by JSPS KAKENHI JP22H05106, JP23K28045, JP26H02495 (to HS), JP26KJ1485 (to MO), JP25K24366 (to HY), JST CREST JPMJCR21N3 (to HS), JST Moonshot JPMJMS263I (to HS), and JST BOOST JPMJBS2407 (to MO).

Ethics Statement

This work uses publicly available datasets (the Pile, uncopyrighted version (Gao et al., 2020), and Common Pile v0.1 (Kandpal et al., 2025)) that have been released for research purposes under open licenses. The 100 language models analyzed in our experiments are all openly available on the Hugging Face Hub. Our framework aims to make comparisons among many LLMs more computationally efficient, which we believe contributes to transparency and reproducibility in language model research.

Use of AI Assistants

We used a generative AI assistant for language polishing, coding support, and paper drafting from experimental codes; all outputs were reviewed and verified by the authors.

References

- Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Hassan Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, and 66 others. 2024a. [Phi-3 technical report: A highly capable language model locally on your phone](#). Technical Report MSR-TR-2024-12, Microsoft.
- Marah I Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yanzhi Li, Weishung Liu, Caio CT Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024b. [Phi-4 technical report](#). Technical Report MSR-TR-2024-57, Microsoft.

- Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgrén, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, and 3 others. 2025. [Smollm2: When smol goes big – data-centric training of a small language model](#). *Preprint*, arXiv:2502.02737.
- Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Leandro von Werra, and Thomas Wolf. 2024. Smollm - blazingly fast and remarkably powerful.
- Frank B. Baker. 2001. *The basics of item response theory*, 2nd ed edition. ERIC Clearinghouse on Assessment and Evaluation, College Park, Md.
- Gabrielle Berrada, Jannik Kossen, Freddie Bickford Smith, Muhammed Razzak, Yarin Gal, and Tom Rainforth. 2025. [Scaling up active testing to large language models](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. [Pythia: A suite for analyzing large language models across training and scaling](#). *Preprint*, arXiv:2304.01373.
- Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. [GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow](#).
- Monica F. Bugallo, Victor Elvira, Luca Martino, David Luengo, Joaquin Miguez, and Petar M. Djuric. 2017. [Adaptive Importance Sampling: The past, the present, and the future](#). *IEEE Signal Processing Magazine*, 34(4):60–79.
- Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre F. T. Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, and Michael Desa. 2024. [Saullm-7b: A pioneering large language model for law](#). *Preprint*, arXiv:2403.03883.
- John Dang, Shivalika Singh, Daniel D’souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi-Chern Tan, Tom Kocmi, Florian Strub, Nathan Grinsztajn, Yannis Flet-Berliac, and 26 others. 2024. [Aya expanse: Combining research breakthroughs for a new multilingual frontier](#). *Preprint*, arXiv:2412.04261.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint*, arXiv:2501.12948.
- DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, and 138 others. 2024. [Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model](#). *Preprint*, arXiv:2405.04434.
- Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. [The Pile: An 800GB Dataset of Diverse Text for Language Modeling](#). *Preprint*, arXiv:2101.00027.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. [SimCSE: Simple contrastive learning of sentence embeddings](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Gemma Team. 2024. [Gemma](#).
- Gemma Team. 2025. [Gemma 3](#).
- Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y. Wu, Y. K. Li, Fuli Luo, Yingfei Xiong, and Wenfeng Liang. 2024. [Deepseek-coder: When the large language model meets programming – the rise of code intelligence](#). *Preprint*, arXiv:2401.14196.
- Yuto Harada, Yusuke Yamauchi, Yusuke Oda, Yohei Oseki, Yusuke Miyao, and Yu Takagi. 2025. [Massive supervised fine-tuning experiments reveal how data, layer, and training factors shape LLM alignment quality](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 22360–22381, Suzhou, China. Association for Computational Linguistics.
- Alejandro Hernández-Cano, Alexander Hägele, Allen Hao Huang, Angelika Romanou, Antoni-Joan Solergibert, Barna Pásztor, Bettina Messmer, Dhia Garbaya, Eduard Frank Ďurech, Ido Hakimi, Juan Garcia Giraldo, Mete Ismayilzada, Negar Foroutan, Skander Moalla, Tiancheng Chen, Vinko Sabolčec, Yixuan Xu, Michael Aerni, Badr AlKhamissi, and 83 others. 2026. [Apertus: Democratizing open and compliant LLMs for global language environments](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 46877–46955, San Diego, California, United States. Association for Computational Linguistics.
- Tim Hesterberg. 1995. [Weighted Average Importance Sampling and Defensive Mixture Distributions](#). *Technometrics*, 37(2):185–194.

- Eliahu Horwitz, Nitzan Kurer, Jonathan Kahana, Liel Amar, and Yedid Hoshen. 2025. [We Should Chart an Atlas of All the World’s Models](#). In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems Position Paper Track*.
- Binyuan Hui, Jian Yang, Zeyu Cui, Jiayi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, and 1 others. 2024. Qwen2.5-coder technical report. *arXiv preprint arXiv:2409.12186*.
- Ghassen Jerfel, Serena Wang, Clara Wong-Fannjiang, Katherine A. Heller, Yian Ma, and Michael I. Jordan. 2021. [Variational refinement for importance sampling using the forward Kullback-Leibler divergence](#). In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Nikhil Kandpal, Brian Lester, Colin Raffel, Sebastian Majstorovic, Stella Biderman, Baber Abbasi, Luca Soldaini, Enrico Shippole, A. Feder Cooper, Aviya Skowron, Shayne Longpre, Lintang Sutawika, Alon Albalak, Zhenlin Xu, Guilherme Penedo, Loubna Ben allal, Elie Bakouch, John David Pressman, Honglu Fan, and 8 others. 2025. [The Common Pile v0.1: An 8TB Dataset of Public Domain and Openly Licensed Text](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Ryo Kishino, Yusuke Takase, Momose Oyama, Hiroaki Yamagiwa, and Hidetoshi Shimodaira. 2026. [Establishing a scale for Kullback-Leibler divergence in language models across various settings](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 23223–23248, San Diego, California, United States. Association for Computational Linguistics.
- Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. 2021. [Active testing: Sample-efficient model evaluation](#). In *Proceedings of the 38th International Conference on Machine Learning*.
- Jannik Kossen, Sebastian Farquhar, Yarin Gal, and Tom Rainforth. 2022. [Active surrogate estimators: an active learning approach to label-efficient model evaluation](#). In *Proceedings of the 36th International Conference on Neural Information Processing Systems*.
- Nikolaus Kriegeskorte, Marieke Mur, and Peter A. Bandettini. 2008. [Representational similarity analysis - connecting the branches of systems neuroscience](#). *Frontiers in Systems Neuroscience*, Volume 2 - 2008.
- LG AI Research. 2024. Exaone 3.5: Series of large language models for real-world use cases. *arXiv preprint arXiv:https://arxiv.org/abs/2412.04862*.
- Liquid AI. 2025. Lfm2 technical report. *arXiv preprint arXiv:2511.23404*.
- Llama Team. 2024. [The Llama 3 Herd of Models](#). *Preprint*, arXiv:2407.21783.
- Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. 2024. [tinyBenchmarks: evaluating LLMs with fewer examples](#). In *Proceedings of the 41st International Conference on Machine Learning*.
- Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, and 1 others. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.
- Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. [Large language diffusion models](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2024. [2 OLMo 2 Furious](#). *Preprint*, arXiv:2501.00656.
- OpenAI. 2025. [gpt-oss-120b & gpt-oss-20b Model Card](#). *Preprint*, arXiv:2508.10925.
- Art B. Owen. 2013. *Monte Carlo theory, methods and examples*. <https://artowen.su.domains/mc/>.
- Momose Oyama, Ryo Kishino, Hiroaki Yamagiwa, and Hidetoshi Shimodaira. 2025a. [Likelihood Variance as Text Importance for Resampling Texts to Map Language Models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 9453–9465, Suzhou, China. Association for Computational Linguistics.
- Momose Oyama, Hiroaki Yamagiwa, Yusuke Takase, and Hidetoshi Shimodaira. 2025b. [Mapping 1,000+ Language Models via the Log-Likelihood Vector](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32983–33038.
- Qwen Team. 2024. [Qwen2.5: A Party of Foundation Models](#).
- Qwen Team. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC² Workshop*.
- Kei Sawada, Tianyu Zhao, Makoto Shing, Kentaro Mitsui, Akio Kaga, Yukiya Hono, Toshiaki Wakatsuki, and Koh Mitsuda. 2024. [Release of pre-trained models for the Japanese language](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13898–13905.
- H. S. Seung, M. Oppen, and H. Sompolinsky. 1992. [Query by Committee](#). In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92*, page 287–294, New York, NY, USA. Association for Computing Machinery.
- Yikang Shen, Matthew Stallone, Mayank Mishra, Gaoyuan Zhang, Shawn Tan, Aditya Prasad, Adriana Meza Soria, David D. Cox, and Rameswar Panda. 2024. [Power scheduler: A batch size and token number agnostic learning rate scheduler](#). *Preprint*, arXiv:2408.13359.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *Preprint*, arXiv:2307.09288.
- Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. 2024. [Anchor points: Benchmarking models with much fewer examples](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1576–1601, St. Julian’s, Malta. Association for Computational Linguistics.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024a. [Qwen2 technical report](#). *Preprint*, arXiv:2407.10671.
- An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024b. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.
- Nicolas Yax, Pierre-Yves Oudeyer, and Stefano Palminteri. 2025. [PhyloLM: Inferring the phylogeny of large language models and predicting their performances in benchmarks](#). In *Proceedings of the 13th International Conference on Learning Representations*.
- Gaifan Zhang and Danushka Bollegala. 2026. [Map of encoders – mapping sentence encoders using quantum relative entropy](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14160–14208, San Diego, California, United States. Association for Computational Linguistics.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, JJ (Jingjing) Liu, and Bill Dolan. 2019. [Dialogpt: Large-scale generative pre-training for conversational response generation](#). In *arXiv:1911.00536*.
- Sally Zhu, Ahmed M Ahmed, Rohith Kudipudi, and Percy Liang. 2025. [Independence tests for language models](#). In *Proceedings of the 42nd International Conference on Machine Learning*.
- Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. 2025. [EmbedLLM: Learning compact representations of large language models](#). In *Proceedings of the 13th International Conference on Learning Representations*.

A Details of Experimental Settings

Models. We used 100 open-source LLMs available on the Hugging Face Hub for the KL divergence estimation experiment. The selection criteria were as follows: (1) Task category is “Text Generation”. (2) Parameter count is 24B or less (an upper bound set by computational and time constraints). (3) Ranked in the top downloads over the last 30 days as of Dec. 13, 2025. To unify the inference environment and ensure fair comparison, quantized models (e.g., AWQ, GGUF, GPTQ, 4-bit/8-bit variants) were excluded. The 100 open-source LLMs used in our experiments are listed in Table 4.

Text Data ($\mathcal{D}^*$). $\mathcal{D}^*$, used for training and validating the predictor, consists of 10,000 texts collected from the Pile (Gao et al., 2020)⁹. We computed the log-likelihoods of all 100 target LLMs on these 10,000 texts and used them to compute the variance $V(x; \mathcal{M})$ that serves as the predictor’s target.

Text Data ($\mathcal{D}_{\text{pile}}^\dagger$). $\mathcal{D}_{\text{pile}}^\dagger$ is constructed by sampling $N_0 = 10^6$ texts from the uncopyrighted version of the Pile¹⁰, disjoint from $\mathcal{D}^*$. Each text is a 1,024-byte chunk.

Text Data (Domain Generalization). For the domain generalization experiment, we use 24 domains from Common Pile v0.1 (Kandpal et al.,

⁹<https://github.com/shimo-lab/modelmap/tree/main/1000models>

¹⁰<https://huggingface.co/datasets/monology/pile-uncopyrighted>

Model Name	Params	Downloads	Model Name	Params	Downloads
Qwen/Qwen2.5-3B-Instruct (Qwen Team, 2024)	3B	9.81M	deepseek-ai/DeepSeek-R1-0528-Qwen3-8B (DeepSeek-AI et al., 2025)	8B	509k
openai-community/gpt2 (Radford et al., 2019)	0.1B	9.2M	microsoft/phi-4 (Abdin et al., 2024b)	15B	479k
openai/gpt-oss-20b (OpenAI, 2025)	22B	8M	Qwen/Qwen2-0.5B (Yang et al., 2024a)	0.5B	469k
meta-llama/Llama-3.1-8B-Instruct (Llama Team, 2024)	8B	7.54M	Qwen/Qwen2.5-Coder-1.5B (Hui et al., 2024)	2B	468k
Qwen/Qwen3-0.6B (Qwen Team, 2025)	0.8B	7.53M	inference-net/Schematron-3B	3B	467k
Qwen/Qwen2.5-7B-Instruct (Qwen Team, 2024)	8B	6.72M	Qwen/Qwen2.5-Math-1.5B (Yang et al., 2024b)	2B	459k
Qwen/Qwen2.5-Coder-0.5B-Instruct (Hui et al., 2024)	0.5B	5.95M	EssentialAI/rnj-1-instruct	8B	447k
Qwen/Qwen2.5-1.5B-Instruct (Qwen Team, 2024)	2B	5.56M	HuggingFaceTB/SmolLM-135M (Allal et al., 2024)	0.1B	411k
Qwen/Qwen3-4B-Instruct-2507 (Qwen Team, 2025)	4B	5.27M	swiss-ai/Apertus-8B-Instruct-2509 (Hernández-Cano et al., 2026)	8B	409k
Qwen/Qwen3-1.7B (Qwen Team, 2025)	2B	4.41M	NousResearch/Llama-2-7b-hf	7B	402k
Qwen/Qwen3-8B (Qwen Team, 2025)	8B	4.32M	IlyaGusev/saiga_llama3_8b	8B	395k
meta-llama/Llama-3.2-1B-Instruct (Llama Team, 2024)	1B	3.47M	microsoft/Phi-3.5-mini-instruct (Abdin et al., 2024a)	4B	387k
mistralai/Mistral-7B-Instruct-v0.2 (Jiang et al., 2023)	7B	3.41M	Qwen/Qwen2.5-14B (Qwen Team, 2024)	15B	382k
meta-llama/Llama-3.2-1B (Llama Team, 2024)	1B	3.08M	mistralai/Mistral-7B-Instruct-v0.1 (Jiang et al., 2023)	7B	372k
Qwen/Qwen3-4B (Qwen Team, 2025)	4B	2.93M	Qwen/Qwen3-4B-Base (Qwen Team, 2025)	4B	372k
deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI et al., 2025)	2B	2.58M	unsloth/gpt-oss-20b-BF16	21B	355k
google/gemma-3-1b-it (Gemma Team, 2025)	1.0B	2.44M	mistralai/Mistral-7B-v0.1 (Jiang et al., 2023)	7B	349k
trl-internal-testing/tiny-Qwen2ForCausalLM-2.5	2.43M	2.38M	NousResearch/Hermes-3-Llama-3.1-8B	8B	347k
distilbert/distilgpt2 (Sanh et al., 2019)	88.2M	2.27M	meta-llama/Llama-3.2-3B (Llama Team, 2024)	3B	343k
Qwen/Qwen2.5-0.5B-Instruct (Qwen Team, 2024)	0.5B	2.26M	Qwen/Qwen3-1.7B-Base (Qwen Team, 2025)	2B	333k
context-labs/meta-llama-Llama-3.2-3B-Instruct-FP16	3B	2.09M	meta-llama/Llama-2-7b-chat-hf (Touvron et al., 2023)	7B	332k
meta-llama/Meta-Llama-3-8B (Llama Team, 2024)	8B	2.09M	ibm-research/PowerMoE-3b (Shen et al., 2024)	3B	316k
microsoft/Phi-3-mini-4k-instruct (Abdin et al., 2024a)	4B	2.04M	OpenPipe/Qwen3-14B-Instruct	15B	309k
TinyLlama/TinyLlama-1.1B-Chat-v1.0	1B	1.8M	microsoft/DialoGPT-small (Zhang et al., 2019)	0.2B	294k
meta-llama/Llama-3.2-3B-Instruct (Llama Team, 2024)	3B	1.69M	HuggingFaceTB/SmolLM2-135M-Instruct(Allal et al., 2025)	0.1B	286k
meta-llama/Meta-Llama-3-8B-Instruct (Llama Team, 2024)	8B	1.37M	llamafactory/tiny-random-Llama-3	4.11M	269k
bigscience/bloomz-560m (Muennighoff et al., 2022)	0.6B	1.34M	meta-llama/Llama-2-13b-chat-hf (Touvron et al., 2023)	13B	269k
deepseek-ai/DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI et al., 2025)	8B	1.31M	Qwen/Qwen2.5-3B (Qwen Team, 2024)	3B	267k
microsoft/phi-2	3B	1.08M	huggyllama/llama-7b	7B	266k
deepseek-ai/DeepSeek-R1-Distill-Llama-8B (DeepSeek-AI et al., 2025)	8B	1.01M	EleutherAI/pythia-70m-deduped (Biderman et al., 2023)	95.6M	256k
Qwen/Qwen2.5-Math-7B (Yang et al., 2024b)	8B	997k	ibm-granite/granite-3.3-2b-instruct	3B	241k
Qwen/Qwen3-14B (Qwen Team, 2025)	15B	939k	deepseek-ai/DeepSeek-V2-Lite-Chat (DeepSeek-AI et al., 2024)	16B	238k
openai-community/gpt2-large (Radford et al., 2019)	0.8B	930k	openai-community/gpt2-xl (Radford et al., 2019)	2B	233k
Qwen/Qwen2.5-7B (Qwen Team, 2024)	8B	903k	trl-internal-testing/tiny-random-LlamaForCausalLM	1.03M	230k
hmlor/tiny-random-LlamaForCausalLM	1.06M	892k	Qwen/Qwen3-8B-Base (Qwen Team, 2025)	8B	228k
Qwen/Qwen2.5-1.5B (Qwen Team, 2024)	2B	831k	GSAI-ML/LLaDA-8B-Instruct (Nie et al., 2025)	8B	223k
openai-community/gpt2-medium (Radford et al., 2019)	0.4B	812k	CoHeresLabs/aya-expense-8b (Dang et al., 2024)	8B	222k
Qwen/Qwen2.5-0.5B (Qwen Team, 2024)	0.5B	769k	deepseek-ai/DeepSeek-Coder-V2-Lite-Instruct (DeepSeek-AI et al., 2024)	16B	221k
Gensyn/Qwen2.5-0.5B-Instruct	0.5B	755k	unsloth/Meta-Llama-3.1-8B-Instruct	8B	214k
google/gemma-2-2b-it (Gemma Team, 2024)	3B	742k	microsoft/Phi-3-mini-128k-instruct (Abdin et al., 2024a)	4B	211k
meta-llama/Llama-3.1-8B (Llama Team, 2024)	8B	729k	google/gemma-2b (Gemma Team, 2024)	3B	205k
Qwen/Qwen2.5-Coder-7B-Instruct (Hui et al., 2024)	8B	677k	EleutherAI/gpt-neo-125m (Black et al., 2021)	0.2B	204k
Qwen/Qwen2.5-14B-Instruct (Qwen Team, 2024)	15B	673k	NousResearch/Meta-Llama-3.1-8B-Instruct	8B	201k
deepseek-ai/deepseek-coder-1.3b-instruct (Guo et al., 2024)	1B	636k	Equal8/Saul-7B-Instruct-v1 (Colombo et al., 2024)	7B	200k
Qwen/Qwen2-1.5B-Instruct (Yang et al., 2024a)	2B	625k	utter-project/EuroLLM-1.7B-Instruct	2B	198k
HuggingFaceTB/SmolLM2-135M (Allal et al., 2025)	0.1B	615k	hmlor/tiny-random-Gemma2ForCausalLM	8.44M	197k
Qwen/Qwen3-4B-Thinking-2507 (Qwen Team, 2025)	4B	608k	Qwen/Qwen2-0.5B-Instruct (Yang et al., 2024a)	0.5B	197k
LiquidAI/LFM2-1.2B (Liquid AI, 2025)	1B	534k	tekium/OpenHermes-2.5-Mistral-7B	7B	195k
meta-llama/Llama-2-7b-hf (Touvron et al., 2023)	7B	531k	rinna/japanese-gpt-neox-small (Sawada et al., 2024)	0.2B	194k
allenai/OLMo-2-0425-1B (OLMo et al., 2024)	1B	512k	LGAI-EXAONE/EXAONE-3.5-2.4B-Instruct (LG AI Research, 2024)	2B	189k

Table 4: List of the 100 models used in the experiments.

2025). Common Pile v0.1 originally consists of 31 domains, but we excluded 7 domains whose total text volume is less than 10GB in order to keep the population size consistent across domains: foodista, libretexts, news, oercommons, pressbooks, public_domain_review, python_enhancement_proposals. For each of the remaining 24 domains, we extracted approximately 10GB of text, split it into 1,024-byte chunks, and randomly sampled $N_0 = 10^6$ chunks to construct $\mathcal{D}_{\text{domain}}^\dagger$. The 24 domains used in our experiments are listed in Table 1.

Computation of Embeddings. The text embeddings used for PredIS and for training the predictor are computed with SimCSE-RoBERTa-base (Gao et al., 2021). For the embedding computation, each text is truncated to 512 tokens, the maximum input length of the embedding model. Computing the embeddings of the $|\mathcal{D}^\dagger| = 10^6$ texts takes about 30 minutes on a single A100 (40GB) GPU.

Computation of Log-Likelihood. The log-likelihood is computed using the autoregressive

factorization

$$\log p(x) = \sum_{i=1}^{|x|} \log p(x_i | x_{<i}).$$

Each text is tokenized into $x = (x_1, \dots, x_{|x|})$ with each model’s own tokenizer. Then, it is passed through the model in a single forward pass. For each position, we apply a log-softmax to the output logits and extract the log-probability assigned to the observed next token. Summing these per-token conditional log-probabilities yields $\log p(x)$ ¹¹. To compute the population-level KL divergence g_{ij} for evaluation, we computed log-likelihoods of all 100 LLMs over each of the 25 domain populations $\mathcal{D}^\dagger$. On a single NVIDIA A100 (40GB) GPU, computing log-likelihoods for $|\mathcal{D}^\dagger| = 10^6$ texts took on average about 3 hours per model.

B Predictor Ablation

This appendix elaborates on the training target of the predictor introduced in Section 4.1, where we mentioned $K = 99$ LLMs.

¹¹Note that $|x|$ denotes the token count under each model’s own tokenizer, which differs from $L(x)$, measured consistently with the GPT-2 tokenizer.

Domain of $\mathcal{D}_{\text{domain}}^{\dagger}$	Mean Rel. RMSE ($\downarrow$)				Median Eff. Gain ($\uparrow$)		Mean RMSE ($\downarrow$)			
	Unif	Length	PredIS-100	PredIS-99	PredIS-100	PredIS-99	Unif	Length	PredIS-100	PredIS-99
the Pile	0.047	0.035	0.043	0.033	1.18	1.80	1.28e+04	7.55e+03	9.35e+03	6.94e+03
arxiv abstracts	0.023	0.023	0.024	0.023	1.00	1.19	5.59e+02	4.11e+02	3.84e+02	4.18e+02
arxiv papers	0.041	0.036	0.037	0.028	0.85	2.00	3.75e+03	2.66e+03	2.74e+03	2.41e+03
biodiversity heritage library	0.027	0.023	0.023	0.022	1.35	1.50	2.87e+02	2.22e+02	2.26e+02	2.06e+02
caselaw access project	0.026	0.024	0.022	0.021	1.29	1.35	3.58e+02	2.90e+02	2.34e+02	2.54e+02
ccccc	0.039	0.032	0.027	0.028	1.84	1.86	2.26e+03	1.46e+03	1.07e+03	1.10e+03
data provenance initiative	0.037	0.031	0.036	0.029	0.91	1.40	4.83e+03	2.58e+03	3.51e+03	2.14e+03
doab	0.050	0.036	0.029	0.030	1.82	2.16	3.50e+03	2.16e+03	2.49e+03	2.02e+03
github archive	0.139	0.258	0.150	0.089	1.06	1.71	6.73e+03	9.44e+03	4.55e+03	3.16e+03
library of congress	0.043	0.038	0.034	0.035	1.62	1.46	3.62e+02	3.41e+02	2.52e+02	2.36e+02
peS2o	0.038	0.026	0.024	0.025	1.47	1.41	8.43e+02	5.36e+02	2.76e+02	3.73e+02
pre 1929 books	0.044	0.038	0.032	0.031	1.39	1.39	1.34e+03	8.06e+02	4.43e+02	6.30e+02
project gutenberg	0.050	0.038	0.038	0.032	1.60	2.12	2.78e+03	1.81e+03	1.57e+03	1.25e+03
pubmed	0.022	0.020	0.032	0.023	0.48	0.92	7.19e+03	5.83e+03	1.00e+04	6.84e+03
regulations	0.064	0.043	0.046	0.041	0.88	1.64	1.51e+04	1.13e+04	1.62e+04	1.20e+04
stackexchange	0.114	0.059	0.124	0.078	1.54	2.09	3.69e+03	2.67e+03	2.77e+03	2.24e+03
stackv2 edu	0.050	0.054	0.055	0.050	1.06	1.15	8.50e+03	7.26e+03	6.51e+03	6.03e+03
stackv2 html	0.042	0.036	0.036	0.034	1.10	1.35	3.71e+03	2.73e+03	2.54e+03	2.03e+03
ubuntu irc	0.058	0.060	0.048	0.050	1.51	1.38	1.39e+03	9.21e+02	7.43e+02	8.10e+02
uk hansard	0.054	0.040	0.029	0.031	3.11	2.22	1.48e+03	1.03e+03	6.95e+02	6.81e+02
usgpo	0.147	0.104	0.071	0.068	1.15	2.08	8.99e+03	5.78e+03	7.48e+03	4.76e+03
uspto	0.058	0.041	0.059	0.038	0.78	2.42	5.23e+03	3.15e+03	6.72e+03	2.96e+03
wikimedia	0.064	0.047	0.040	0.040	1.39	1.67	3.08e+03	1.99e+03	1.55e+03	1.60e+03
wikiteam	0.064	0.051	0.047	0.044	1.42	1.73	2.99e+03	2.10e+03	1.87e+03	1.65e+03
youtube	0.211	0.179	0.060	0.103	9.67	4.46	2.88e+03	1.91e+03	1.08e+03	1.27e+03

Table 5: **Predictor ablation: including (PredIS-100) vs. excluding (PredIS-99) the outlier model DialoGPT-small.** Results over $T = 100$ trials at $N = 10,000$, with $\tau = 0.5$, $\alpha = 0.1$. Both perform comparably overall, with PredIS-99 slightly more stable across domains. Best non-Uniform value per metric is in **bold**.

Domain of $\mathcal{D}_{\text{domain}}^{\dagger}$	$N = 100$				$N = 1000$				$N = 10000$			
	Unif	Length	PredIS	Oracle	Unif	Length	PredIS	Oracle	Unif	Length	PredIS	Oracle
the Pile	0.978 $\pm$ 0.013	0.983 $\pm$ 0.010	0.985 $\pm$ 0.009	0.986 $\pm$ 0.006	0.996 $\pm$ 0.003	0.997 $\pm$ 0.002	0.998 $\pm$ 0.001	0.998 $\pm$ 0.001	0.999 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
arxiv abstracts	0.984 $\pm$ 0.007	0.986 $\pm$ 0.005	0.986 $\pm$ 0.006	0.988 $\pm$ 0.004	0.998 $\pm$ 0.001	0.998 $\pm$ 0.001	0.998 $\pm$ 0.001	0.999 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
arxiv papers	0.981 $\pm$ 0.021	0.983 $\pm$ 0.009	0.987 $\pm$ 0.009	0.987 $\pm$ 0.007	0.997 $\pm$ 0.005	0.997 $\pm$ 0.004	0.998 $\pm$ 0.002	0.998 $\pm$ 0.001	0.999 $\pm$ 0.001	1.000 $\pm$ 0.001	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
biodiversity heritage library	0.980 $\pm$ 0.009	0.982 $\pm$ 0.009	0.983 $\pm$ 0.007	0.985 $\pm$ 0.005	0.998 $\pm$ 0.002	0.998 $\pm$ 0.002	0.998 $\pm$ 0.001	0.998 $\pm$ 0.001	1.000 $\pm$ 0.001	1.000 $\pm$ 0.001	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
caselaw access project	0.984 $\pm$ 0.006	0.987 $\pm$ 0.005	0.988 $\pm$ 0.004	0.989 $\pm$ 0.004	0.998 $\pm$ 0.001	0.998 $\pm$ 0.001	0.999 $\pm$ 0.001	0.999 $\pm$ 0.001	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
ccccc	0.966 $\pm$ 0.019	0.970 $\pm$ 0.014	0.972 $\pm$ 0.011	0.976 $\pm$ 0.008	0.995 $\pm$ 0.003	0.996 $\pm$ 0.002	0.996 $\pm$ 0.002	0.997 $\pm$ 0.001	0.999 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
data provenance initiative	0.976 $\pm$ 0.010	0.981 $\pm$ 0.008	0.979 $\pm$ 0.009	0.984 $\pm$ 0.007	0.997 $\pm$ 0.003	0.997 $\pm$ 0.002	0.997 $\pm$ 0.001	0.998 $\pm$ 0.001	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
doab	0.976 $\pm$ 0.022	0.979 $\pm$ 0.015	0.982 $\pm$ 0.013	0.983 $\pm$ 0.007	0.993 $\pm$ 0.008	0.995 $\pm$ 0.003	0.996 $\pm$ 0.002	0.997 $\pm$ 0.001	0.999 $\pm$ 0.001	0.999 $\pm$ 0.000	0.999 $\pm$ 0.000	1.000 $\pm$ 0.000
github archive	0.966 $\pm$ 0.027	0.972 $\pm$ 0.019	0.972 $\pm$ 0.024	0.971 $\pm$ 0.018	0.984 $\pm$ 0.020	0.986 $\pm$ 0.027	0.987 $\pm$ 0.019	0.992 $\pm$ 0.004	0.990 $\pm$ 0.006	0.987 $\pm$ 0.016	0.992 $\pm$ 0.004	0.999 $\pm$ 0.001
library of congress	0.959 $\pm$ 0.018	0.963 $\pm$ 0.016	0.966 $\pm$ 0.014	0.977 $\pm$ 0.008	0.994 $\pm$ 0.004	0.994 $\pm$ 0.003	0.995 $\pm$ 0.002	0.997 $\pm$ 0.001	0.999 $\pm$ 0.000	0.999 $\pm$ 0.000	0.999 $\pm$ 0.000	1.000 $\pm$ 0.000
peS2o	0.985 $\pm$ 0.011	0.987 $\pm$ 0.008	0.986 $\pm$ 0.008	0.988 $\pm$ 0.006	0.998 $\pm$ 0.003	0.998 $\pm$ 0.002	0.998 $\pm$ 0.001	0.998 $\pm$ 0.001	1.000 $\pm$ 0.001	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
pre 1929 books	0.966 $\pm$ 0.023	0.970 $\pm$ 0.015	0.972 $\pm$ 0.014	0.981 $\pm$ 0.008	0.995 $\pm$ 0.003	0.996 $\pm$ 0.002	0.996 $\pm$ 0.002	0.998 $\pm$ 0.001	0.999 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
project gutenberg	0.974 $\pm$ 0.014	0.979 $\pm$ 0.011	0.982 $\pm$ 0.009	0.985 $\pm$ 0.005	0.996 $\pm$ 0.002	0.997 $\pm$ 0.001	0.997 $\pm$ 0.001	0.998 $\pm$ 0.001	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
pubmed	0.994 $\pm$ 0.004	0.995 $\pm$ 0.002	0.994 $\pm$ 0.003	0.995 $\pm$ 0.002	0.999 $\pm$ 0.000	0.999 $\pm$ 0.001	0.999 $\pm$ 0.000	0.999 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
regulations	0.989 $\pm$ 0.013	0.991 $\pm$ 0.007	0.989 $\pm$ 0.008	0.990 $\pm$ 0.007	0.996 $\pm$ 0.005	0.997 $\pm$ 0.004	0.998 $\pm$ 0.002	0.998 $\pm$ 0.002	0.999 $\pm$ 0.001	0.999 $\pm$ 0.001	1.000 $\pm$ 0.001	1.000 $\pm$ 0.000
stackexchange	0.986 $\pm$ 0.011	0.987 $\pm$ 0.017	0.987 $\pm$ 0.016	0.987 $\pm$ 0.017	0.994 $\pm$ 0.017	0.996 $\pm$ 0.014	0.995 $\pm$ 0.016	0.996 $\pm$ 0.004	0.996 $\pm$ 0.008	0.997 $\pm$ 0.012	0.997 $\pm$ 0.005	0.999 $\pm$ 0.000
stackv2 edu	0.977 $\pm$ 0.013	0.977 $\pm$ 0.012	0.974 $\pm$ 0.013	0.978 $\pm$ 0.010	0.995 $\pm$ 0.005	0.995 $\pm$ 0.005	0.995 $\pm$ 0.005	0.996 $\pm$ 0.003	0.999 $\pm$ 0.002	0.999 $\pm$ 0.002	0.999 $\pm$ 0.002	0.999 $\pm$ 0.000
stackv2 html	0.967 $\pm$ 0.017	0.971 $\pm$ 0.015	0.971 $\pm$ 0.015	0.977 $\pm$ 0.009	0.994 $\pm$ 0.005	0.996 $\pm$ 0.003	0.996 $\pm$ 0.002	0.997 $\pm$ 0.001	0.999 $\pm$ 0.001	0.999 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
ubuntu irc	0.965 $\pm$ 0.019	0.965 $\pm$ 0.016	0.966 $\pm$ 0.018	0.975 $\pm$ 0.016	0.994 $\pm$ 0.013	0.994 $\pm$ 0.013	0.994 $\pm$ 0.009	0.997 $\pm$ 0.002	0.998 $\pm$ 0.003	0.999 $\pm$ 0.003	0.999 $\pm$ 0.001	1.000 $\pm$ 0.000
uk hansard	0.950 $\pm$ 0.040	0.971 $\pm$ 0.028	0.984 $\pm$ 0.016	0.989 $\pm$ 0.009	0.996 $\pm$ 0.003	0.998 $\pm$ 0.002	0.998 $\pm$ 0.001	0.999 $\pm$ 0.001	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
usgpo	0.972 $\pm$ 0.024	0.973 $\pm$ 0.022	0.971 $\pm$ 0.034	0.973 $\pm$ 0.024	0.983 $\pm$ 0.020	0.985 $\pm$ 0.016	0.989 $\pm$ 0.015	0.993 $\pm$ 0.007	0.991 $\pm$ 0.007	0.996 $\pm$ 0.004	0.998 $\pm$ 0.002	0.999 $\pm$ 0.001
uspto	0.968 $\pm$ 0.027	0.975 $\pm$ 0.023	0.977 $\pm$ 0.019	0.980 $\pm$ 0.013	0.993 $\pm$ 0.010	0.996 $\pm$ 0.004	0.996 $\pm$ 0.003	0.997 $\pm$ 0.002	0.999 $\pm$ 0.001	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000	1.000 $\pm$ 0.000
wikimedia	0.979 $\pm$ 0.010	0.981 $\pm$ 0.015	0.981 $\pm$ 0.012	0.983 $\pm$ 0.012	0.995 $\pm$ 0.009	0.995 $\pm$ 0.009	0.996 $\pm$ 0.008	0.997 $\pm$ 0.002	0.998 $\pm$ 0.002	0.999 $\pm$ 0.002	0.999 $\pm$ 0.001	1.000 $\pm$ 0.000
wikiteam	0.978 $\pm$ 0.020	0.982 $\pm$ 0.012	0.983 $\pm$ 0.015	0.984 $\pm$ 0.012	0.994 $\pm$ 0.014	0.996 $\pm$ 0.011	0.996 $\pm$ 0.012	0.997 $\pm$ 0.002	0.998 $\pm$ 0.003	0.999 $\pm$ 0.003	0.999 $\pm$ 0.002	1.000 $\pm$ 0.000
youtube	0.955 $\pm$ 0.018	0.956 $\pm$ 0.035	0.955 $\pm$ 0.041	0.957 $\pm$ 0.037	0.979 $\pm$ 0.037	0.976 $\pm$ 0.040	0.975 $\pm$ 0.030	0.985 $\pm$ 0.008	0.980 $\pm$ 0.025	0.982 $\pm$ 0.014	0.992 $\pm$ 0.006	0.998 $\pm$ 0.001

Table 6: **Spearman correlation between estimated and oracle pairwise KL matrices, for $N \in \{100, 1,000, 10,000\}$.** Mean $\pm$ std over $T = 1,000$ trials. **Bold**: best among Unif, Length and PredIS.

Setup. Let $\mathcal{M}_{100}$ denote the full set of $K = 100$ target LLMs. We examined the norms $\|\xi_i\|$ of the centered log-likelihood vectors $\xi_i \in \mathbb{R}^{|\mathcal{D}_{\text{train}}^*|}$ on $\mathcal{D}_{\text{train}}^*$, and found that DialoGPT-small was an outlier whose norm deviates substantially from the others. We denote the set excluding this model by $\mathcal{M}_{99}$. The $\mathcal{M}$ in the main text refers to $\mathcal{M}_{99}$. In this appendix, we compare two predictors: one trained to predict the variance over the full set $V(x; \mathcal{M}_{100})$ (hereafter PredIS-100), and the other trained to predict $V(x; \mathcal{M}_{99})$ (PredIS-99).

Results. Table 5 reports the results across all 25 domains ($T = 100$ trials, $N = 10,000$, $\tau = 0.5$, $\alpha = 0.1$). Overall, the two predictors perform com-

parably, but PredIS-99 is slightly more accurate. We therefore use PredIS-99 in the main text.

C Implementation of Importance Sampling

Two

duce two adjustments: **(1) Smoothing.** We make the proposal distribution proportional to $\hat{V}(x)^\tau$ with $\tau \in (0, 1]$ to mitigate this concentration. A smaller τ flattens the distribution. **(2) Defensive importance sampling** (Hesterberg, 1995). We mix in the uniform distribution $1/|\mathcal{D}^\dagger|$ with a coefficient $\alpha \in [0, 1]$, ensuring a lower bound of $\pi(x) \geq \alpha/|\mathcal{D}^\dagger|$ for every text. This caps the maximum value of $w(x)$ and stabilizes the estimator. Combining the two adjustments yields the final proposal distribution defined in Section 4.2:

$$\pi^{\text{PredIS}}(x) = (1 - \alpha) \frac{\hat{V}(x)^\tau}{\sum_{x'} \hat{V}(x')^\tau} + \frac{\alpha}{|\mathcal{D}^\dagger|}.$$

Hyperparameter choice. We adopt empirical values $\tau = 0.5$ and $\alpha = 0.1$ without optimization. Potential performance improvements from adaptively tuning τ and α are left for future work.

Note on the Length baseline. The Length baseline uses $\pi^{\text{Length}}(x) \propto L(x)$ without modification (i.e., without smoothing or uniform mixture). This is because, in a preliminary experiment, applying the same adjustments (τ, α) to the Length baseline degraded performance. Since our texts are at most 1,024 bytes, the token count $L(x)$ has a narrower dynamic range than $\hat{V}(x)$, making smoothing and defensive mixture unnecessary.

D Results for Additional Sample Sizes

Table 6 extends the Spearman correlation analysis in the main text to $N \in \{100, 1,000, 10,000\}$, complementing the $N = 10$ results in Table 1. As expected, all methods improve with larger N . PredIS achieves the highest or tied correlation in the majority of domains at each N .