Do we really have to filter out random noise in pre-training data for language models?

Anonymous ACL submission

Abstract

Web-scale pre-training datasets are the cornerstone of LLMs' success. However, text data curated from the internet inevitably contains random noise caused by decoding errors or unregulated web content. In contrast to previous works that focus on low quality or synthetic data, our study **provides the first systematic investigation into such random noise through a cohesive "What-Why-How" framework.** Surprisingly, we observed that the resulting increase in next-token prediction (NTP) loss was significantly lower than the proportion of random noise. We provide a theoretical justification for this phenomenon, which also elucidates the success of multilingual models. On the other hand, experiments show that the model's performance in downstream tasks is not based solely on the NTP loss, which means that random noise may result in degraded downstream performance. To address the potential adverse effects, we introduce a novel plug-and-play Local Gradient Matching loss, which explicitly enhances the denoising capability of the downstream task head by aligning the gradient of normal and perturbed features without requiring knowledge of the model's parameters. Additional experiments on 8 language and 14 vision benchmarks further validate its effectiveness.

1 Introduction

Large language models (LLMs), particularly the GPT series (Radford et al., 2019; Brown, 2020; OpenAI, 2023), have fundamentally transformed the research landscape in natural language processing. The remarkable performance of these autoregressive models is largely attributed to pre-training on extensive datasets, which are gathered by crawling text from the whole internet. Given the sheer volume of these datasets, they inevitably encompass a wide variety of noise (Longpre et al., 2024; Elazar et al., 2024). Consequently, it is imperative to understand its impact, as the quality of pre-training data plays a decisive role in the effectiveness of LLMs (Touvron et al., 2023). Allen-Zhu and Li (2024a); Xie et al. (2023b) highlight that low-quality data can significantly diminish a model's knowledge capacity and performance. Shumailov et al. (2024); Seddik et al. (2024) demonstrate that recursively training LLMs with synthetic data can lead to model collapse.

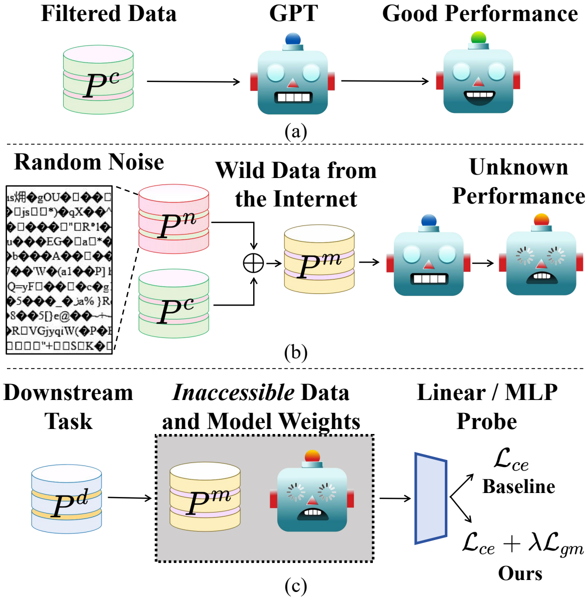


Figure 1: Overview of the study and methodology. (a) The common scenario in which a GPT model, pre-trained on filtered data P^c, demonstrates robust performance. (b) When the pre-training dataset is contaminated with random noise P^n, the resultant language model may exhibit unpredictable behavior. (c) Our approach focuses on the effective fine-tuning of black-box noisy models for downstream tasks P^d.

However, **little attention has been paid to the impact of random noise within datasets**. Due to anti-crawling mechanisms (Gao et al., 2023), decoding errors[1], and tremendous amounts of unmaintained websites[2], the raw data obtained through

[1] https://stackoverflow.com/questions/62499600/gibberish-text-output-because-of-encoding-in-web-scraping

[2] https://community.cloudflare.com/t/website-showing-garbage-text

web crawling inevitably contains a substantial amount of random noise (Zhou et al., 2024; Chen et al., 2022; Kang et al., 2023). Although theoretically it may not be challenging to remove such noise, practical limitations in computational resources often result in incomplete data cleaning(Albalak et al., 2024; Soldaini et al., 2024). For example, it is observed that the Chinese corpus used to train the GPT-4o tokenizer contains a considerable amount of nonsensical data. [3] Therefore, it is of great importance to gain a thorough understanding of the effects of such random noise.

We conduct extensive experiments based on the OpenWebText dataset (Gokaslan et al., 2019) used to pre-train language models with the same architecture and parameter size as GPT-2. Specifically, to simulate random noise shown in Figure 1, we randomly generate, with proportions of 1%, 5% and 20%, a sequence of integers within the range of 0 to 50256, according to the vocabulary size of GPT-2's tokenizer, to simulate the tokenization outcome of nonsensical text found on the internet. Interestingly, we observe that the presence of random noise does not lead to a catastrophic failure in model training; instead, **its effect on autoregressive loss is disproportionately small**, e.g., the increase in loss is only about 1% even with 20% of the dataset being noisy. We provide a theoretical analysis to explain these phenomena, which also sheds light on the success of multilingual models (where one language may appear as "noise" to another) and large speech models (Chen et al., 2022), indicating the broader implications of studying the effects of random noise.

On the other hand, further experiments reveal that a model that exhibits a lower NTP loss experiences a 1.5% decrease in accuracy on downstream tasks. This indicates that performance on downstream tasks is not solely rely upon the NTP loss. Given the common practice of fine-tuning a pre-trained foundation model rather than undergoing a full pre-training process from scratch, we opt to follow the work of Chen et al. (2024) by exploring how to efficiently fine-tune language models only using extracted features for downstream tasks when the pre-training data and model weights are not accessible, which reflects real-world application scenarios for LLMs. To mitigate the potential adverse effects of noise, we propose a novel plug-and-play Local Gradient Matching (LGM) loss. This method involves artificially adding noise to the output features and minimizing the gradient difference between the noisy and original features. We also provide a theoretical analysis of the LGM loss. Interestingly, when applying the LGM loss to fine-tune clean models such as Llama-3 or ViT-L on 8 language and 14 vision datasets, we observe an unexpected improvement in accuracy, which effectively demonstrates the versatility and broad applicability of the LGM loss beyond its original intent of addressing noise-related issues.

The remaining part is arranged as follows. In Section 2, we summarize related works. In Section 3 and 4, we follow a logical structure based on **"What-Why-How"**:

- What: What is the effect of random noise? In Section 3, we demonstrate through experiments how random noise impacts NTP loss.
- Why: Why does it have this effect? The section further explores the underlying reasons by providing a rigorous theoretical analysis.
- How: How do we mitigate the potentially harmful effect on downstream tasks? In Section 4, We introduce the LGM loss and provide a theoretical explanation.

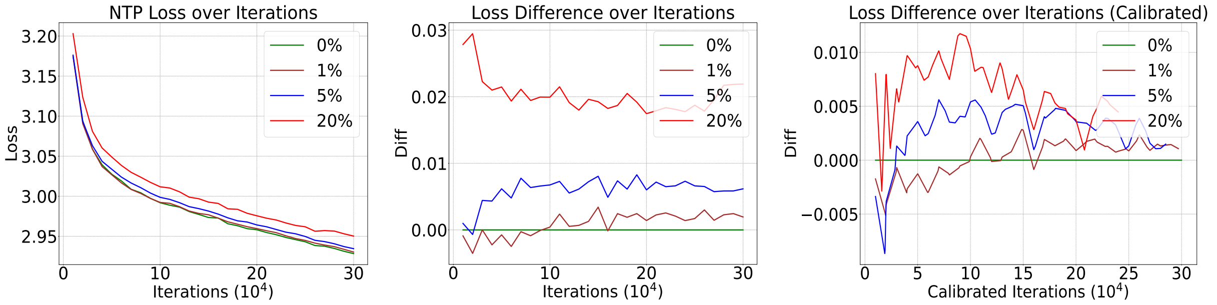
Section 4 further provides evidence of the novelty and effectiveness of the LGM loss by conducting extensive experiments on 22 downstream tasks.

In summary, our contributions are as follows: (1) We investigate the underexplored problem of random noise in pre-training datasets for language models. (2) We pre-train multiple GPT-2 models and the empirical results show that the influence of random noise on NTP loss is insignificant. Then we provide a theoretical analysis, extending our findings to other domains and thus highlighting the significance of this research direction. (3) We propose a novel blackbox fine-tuning LGM loss for downstream tasks, supported by comprehensive experimental and theoretical analysis that confirm its efficacy. **Code, data, and model checkpoint weights are available at** this repo.

2 Related Works

Pre-training Data Analysis for Language Model Training. Elazar et al. (2024) analyzed open-source datasets like The Pile and C4, uncovering significant amounts of low-quality content in these datasets. Allen-Zhu and Li (2024a); Seddik et al. (2024) highlighted the negative impact of such data

[3] https://github.com/jiangyy/gpt-tokens



(a) (b) (c)

Figure 2: Next-token prediction loss on the clean OpenWebText validation set for GPT-2 models pre-trained on synthetic OpenWebText datasets with varying levels of random noise. (a) Trend of NTP loss as training proceeds. (b) Difference in NTP loss between the noisy and clean models after the same number of training iterations. (c) Difference in loss values after undergoing the same number of training iterations on clean OpenWebText data.

on training. Despite these remarkable contributions, there remains a lack of understanding regarding the specific effects of random noise on language model performance. This paper aims to address this gap.

Noisy Model Learning. Our work draws significant inspiration from Noisy Model Learning (NML) proposed by Chen et al. (2024). In NML, the authors introduce noise into large datasets like ImageNet by randomly altering labels, then pretrain neural networks on these noisy datasets. The study reveals that moderate label noise enhances in-distribution (ID) sample classification, while out-of-distribution (OOD) performance deteriorates with increasing noise. This paper extends the concept of NML, presenting theoretical insights and methodologies that are applicable across multiple modalities and various problems.

Due to space limitations, the detailed related works are provided in Appendix C.

3 Revealing the Effect of Random Noise in Language Model Pre-training

In this section, we first pre-train multiple GPT-2 models on synthetic noisy OpenWebText corpus to investigate the impact of random noise in the pre-training data. We then provide a theoretical analysis of the results and validate our theory through experiments. Finally, we demonstrate that the insights gained from our investigation have broader applicability beyond the immediate scope of our study. In summary, Section 3 focuses on understanding the impact of random noise by presenting detailed experiments and theoretical analyses. We delve into how random noise affects NTP loss and introduce insights into why these effects occur. It sets the foundation for our investigation.

The frequently used notation and their descriptions are shown in Appendix A.

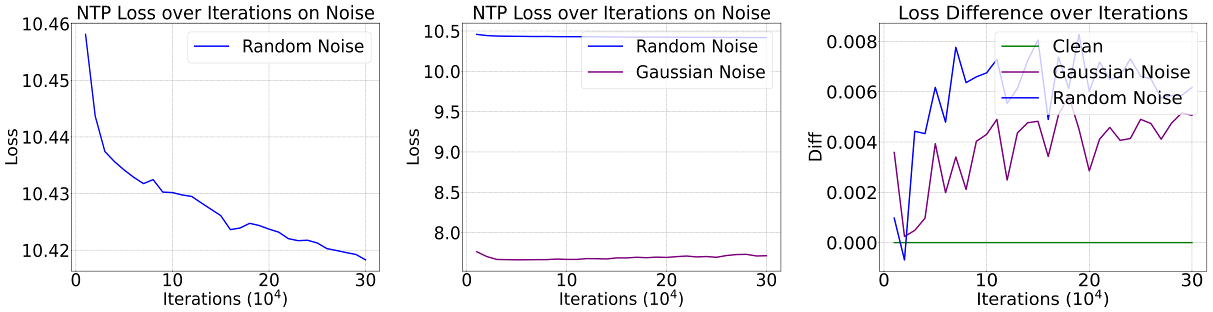
3.1 Experimental Design

Preliminary. Let L denote the maximum context length of the language model and let $\mathcal{W}$ represent the model's vocabulary with size $V = |\mathcal{W}|$. We define $\mathcal{X}$ as the set of all discrete sentences that the model can represent, where $\mathcal{X} = \cup_{i=1}^{L}\{0, 1, \ldots, V-1\}^i = \cup_{i=1}^{L}\mathcal{W}^i$ and $\{0, 1, \ldots, V-1\}^i$ represents prefixes of length i. For any discrete set A, let Δ_A denote the set of all probability distributions defined on A. Given that next-token prediction (NTP) is actually a classification task given the prefix, we define joint probability distributions $P^c, P^n, P^m \in \Delta_{\mathcal{X}\times\mathcal{W}}$ where P^c represents the distribution of clean data, P^n represents the distribution of noise data, and P^m represents the distribution of the mixed pre-training dataset which contains both clean and noise data. Since the noisy dataset can be viewed as the concatenation of clean data and random noise, it can be formalized by the Huber contamination model (Fang et al., 2022) as follows:

$$P^m = \alpha P^n + (1-\alpha)P^c \tag{1}$$

where we use α to represent the noise proportion. An explanation of Equation (1) can be found in Appendix B.1. For any joint probability distribution $P \in \Delta_{\mathcal{X}\times\mathcal{W}}$, let $P_X \in \Delta_{\mathcal{X}}$ and $P_{\cdot|X} \in \Delta_{\mathcal{W}}$ represent the marginal and conditional distribution of P over $\mathcal{X}$ and $\mathcal{W}$.

We use $\mathcal{H}$ to denote the hypothesis space (e.g., all possible parameter configurations given the transformer architecture). Define $h : \mathcal{X} \to \mathbb{R}^V \in \mathcal{H}$ as the language model and $\boldsymbol{p}^h_{\cdot|x}(w)$ as the w-th component of the probability distribution induced by $h(x)$. The next-token prediction loss can be



(a) (b) (c)

Figure 3: Validation experiments. (a) Loss trends on the random noise in the *training set* of the model trained on the dataset with 5% random noise. (b) Comparison of the loss between 5% random noise and Gaussian noise. (c) The loss difference on the clean OpenWebText validation set compared to the baseline for models trained on datasets with 5% random noise and 5% Gaussian noise, respectively.

expressed as follows:

$$\mathcal{L}_{ntp}(P,h) = \mathbb{E}_{x\sim P_X}\mathbb{E}_{w\sim P_{\cdot|x}}\left[-\log(\boldsymbol{p}^h_{\cdot|x}(w))\right]. \tag{2}$$

Data setup. We utilize the OpenWebText dataset (Gokaslan et al., 2019) which comprises 8 billion tokens as an alternative to the original WebText dataset used to train GPT-2 124M models (Radford et al., 2019). Concretely, to mimic the unpredictable nature of garbage noise after it has been tokenized, we first generate a sequence of integers where each integer follows a uniform distribution over $[0, 50256)$ (recall that 50256 is the vocabulary size of GPT-2's tokenizer). Furthermore, we notice that beyond uniform distribution, certain tokens appeare with significantly higher frequency in real-world random noise. To reflect this phenomenon, approximately 100 tokens in $[0, 50256)$ are randomly selected and their frequencies increase accordingly. The overall distribution of the data can be referred to in Figure 8. The generated noise is then added to the clean dataset such that α is 1%, 5%, and 20% respectively. Each synthetic noisy dataset is used to pre-train a GPT-2 model. We argue that given the vast diversity of websites on the internet, it is inevitable that datasets used for training large models will inadvertently include such garbled text as shown in Figure 1. This kind of noise is not just theoretical; it is a practical issue encountered during web crawling and data collection processes.

We set the context length L to be 1024 and the batch size to be 640. All models are trained for 300,000 iterations. To evaluate the performance, the resulting model checkpoints are tested on the clean OpenWebText validation set, measuring the NTP loss for comparison. Further details regarding datasets and experimental parameters can be found in Appendix D.

3.2 Results

In Figure 2, we illustrate the evolution of the NTP loss throughout the training process. Although random noise has a negative effect on the model's performance as expected, experimental results yield two intriguing insights:

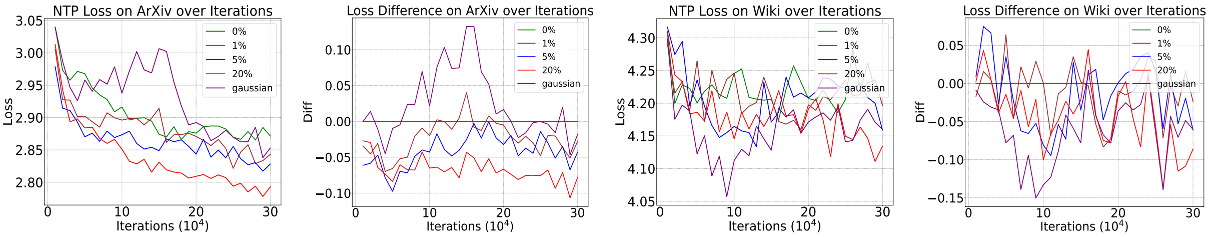
(1) In contrast to the low-quality or synthetic data, **the presence of random noise does not lead to training collapse**, even when the noise level reaches 20%. While increasing training time on low quality or synthetic data typically degrades model performance (Allen-Zhu and Li, 2024a; Shumailov et al., 2024), extending the training duration continues to drive down the model's loss in the case of random noise.

(2) **The impact of random noise on the loss is disproportionately small**. For instance, 5% of random noise only results in a 0.2% increase in the NTP loss. This discrepancy becomes even smaller if the noisy models are calibrated to match the number of training iterations with the baselines trained on clean datasets.

These positive experimental outcomes further corroborate the robustness of language models and provide insights into why pre-training on large-scale datasets that inevitably contain significant amounts of noise can still yield high-performing models. These somewhat unexpected findings naturally prompt us to explore the underlying reasons.

3.3 Theoretical Analysis

In the analysis below, we focus on the impact of random noise on NTP loss, as pre-training loss is crucial for the performance on downstream tasks (Saunshi et al., 2021; Wei et al., 2021; Liu et al., 2023; Zheng et al., 2023a). Specifically, we are



(a) (b) (c) (d)

Figure 4: Loss and its difference across different types and levels of noise within the ArXiv and Wikipedia corpora.

interested in the difference of NTP Loss between a model h^* trained on a noise-free dataset and a model h trained with a noisy dataset. We begin by noting that sampling from the clean distribution should not yield random gibberish and vice versa. Mathematically, this implies that for any prefix r sampled from P^n_X, the probability under the clean distribution $P^c_X(r)$ is zero. Thus, we make the following assumption:

Assumption 1. *P^c and P^n have disjoint support sets, i.e., supp$(P^c) \cap$ supp$(P^n) = \emptyset$.*

The subsequent proposition demonstrates that the error ϵ introduced to the loss due to random noise is less than the proportion α of random noise in the dataset.

Proposition 1. *Under Assumption 1, let h^* be a model trained on P^c, with $\mathcal{L}_{ntp}(P^c, h^*) = -\log p_c$ and $\mathcal{L}_{ntp}(P^n, h^*) = -\log p_n$. When the model h is trained on a mixed distribution P^m which includes noise, it attempts to fit P^n, leading to an increase in the loss on the clean distribution P^c, such that $\mathcal{L}_{ntp}(P^c, h) = -\log(p_c - \epsilon)$ and $\mathcal{L}_{ntp}(P^n, h) = -\log(p_n + \epsilon/k)$ for some $\epsilon > 0$ (k can be shown to be $\Omega(e^{\mathcal{L}_{ntp}(P^n,h)})$). Let $\eta = \alpha p_c - (1-\alpha)kp_n$. We arrive at the following conclusions:*

(1) If $\alpha \leq \frac{kp_n}{p_c + kp_n}$, then for any $0 < \epsilon < p_c$, we have $\mathcal{L}_{ntp}(P^m, h) \geq \mathcal{L}_{ntp}(P^m, h^*)$. This means that when α is sufficiently small, the global minimum on P^m will not be affected by noise.

(2) If $\alpha > \frac{kp_n}{p_c + kp_n}$, then for $\epsilon < \eta$, it holds that $\mathcal{L}_{ntp}(P^m, h) < \mathcal{L}_{ntp}(P^m, h^*)$. This suggests that if α is large enough, the impact on the optimal hypothesis is at least as much as $\alpha p_c - (1-\alpha)kp_n$.

(3) When $\alpha < \frac{1}{3}$ and $k > \frac{\alpha(1-3\alpha)p_c}{(1-\alpha)(2-3\alpha)p_n}$, for $\epsilon \geq 3\eta$ we get $\mathcal{L}_{ntp}(P^m, h^*) < \mathcal{L}_{ntp}(P^m, h)$. Similarly, it can be shown that ϵ does not exceed 2η when $\alpha > \max\left(\frac{kp_n}{p_c + kp_n}, \frac{1}{2}\right)$ and $k > \frac{(2\alpha-1)p_c}{2(1-\alpha)p_n}$. This indicates that when k is sufficiently large, the effect of noise is at most $\mathcal{O}(\alpha p_c - (1-\alpha)kp_n)$.

The proof can be found in Appendix B.2. Proposition 1 primarily investigates the performance gap between models trained on P^m and those on P^c. It is proved that when α is small enough, the presence of noise has no impact on the optimal model on P^m. Even as α approaches $\frac{1}{3}$ or even $\frac{1}{2}$, as long as k is large enough (the analysis regarding k and other parameters is detailed in Appendix B.3), the loss induced by noise, ϵ, does not exceed $\mathcal{O}(\alpha p_c - (1-\alpha)kp_n)$. Given that k is much greater than 1, this implies ϵ is much smaller than αp_c. This explains the observed experimental results.

With these theoretical results in hand, we then conduct multiple experiments to substantiate their validity. First, we plot the trend of NTP loss on random noise within the training set throughout the learning process, as shown in Figure 3(a). It is evident that the loss on random noise decreases at a very slow rate, indicating that the model struggles to efficiently learn the distribution of random noise.

Next, we add 5% Gaussian-distributed noise to the training dataset and compare the results with models trained on 5% random noise. Specifically, we replace the uniform distribution mentioned above with a Gaussian distribution characterized by a mean of 25128 and standard deviation of 500. As depicted in Figure 3(b), the loss on Gaussian noise is lower than that on the random noise. According to Proposition 1, since the Gaussian distribution corresponds to a high p_n, we can **predict** that a model trained on Gaussian noise will exhibit a lower loss on P^c. Figure 3(c) confirms our prediction, thus further validating the proportions.

3.4 Experiments on Other Text Corpus

To further investigate the impact of random noise on model generalization, we evaluate the next-token prediction loss of the trained models on data crawled from arXiv and Wikipedia. The results are illustrated in Figure 4. Surprisingly, models trained with added noise outperformed those trained on P^c. This counterintuitive finding aligns with previous

	SST-2		SST-fine		20newsgroup		CR		Avg	
	Linear	MLP	Linear	MLP	Linear	MLP	Linear	MLP	Linear	MLP
OpenAI's GPT-2*	87.4	/	49.2	/	63.7	/	86.8	/	71.75	/
0%	86.71 ± 0.85	87.36 ± 0.33	49.19 ± 0.32	49.18 ± 0.02	63.12 ± 0.37	62.70 ± 0.86	85.65 ± 0.88	84.86 ± 0.36	71.16	71.02
0% + $\mathcal{L}_{gm}$	**87.42 ± 0.73**	**87.86 ± 0.04**	**49.72 ± 0.27**	**49.81 ± 0.97**	**63.69 ± 0.59**	**62.95 ± 0.13**	**86.58 ± 0.22**	**86.45 ± 0.73**	**71.85**	**71.76**
1%	87.25 ± 0.79	**87.53 ± 0.27**	49.32 ± 0.72	49.45 ± 0.56	63.71 ± 0.02	64.65 ± 0.06	84.86 ± 0.98	84.59 ± 0.59	71.28	71.55
1% + $\mathcal{L}_{gm}$	**87.64 ± 0.91**	87.25 ± 0.44	**49.59 ± 0.73**	**50.01 ± 0.05**	**63.92 ± 0.65**	**64.72 ± 0.76**	**85.12 ± 0.07**	**85.25 ± 0.29**	**71.56**	**71.80**
5%	86.92 ± 0.98	87.23 ± 0.41	49.04 ± 0.11	**50.09 ± 0.53**	63.27 ± 0.79	62.09 ± 0.28	85.30 ± 0.63	**84.32 ± 0.78**	71.13	**70.93**
5% + $\mathcal{L}_{gm}$	**87.19 ± 1.02**	**87.61 ± 0.51**	**49.82 ± 0.17**	48.95 ± 0.89	**63.78 ± 0.93**	**62.37 ± 0.56**	**85.57 ± 0.43**	84.19 ± 0.69	**71.59**	70.78
20%	86.60 ± 1.28	86.60 ± 0.81	49.45 ± 0.78	49.63 ± 0.01	63.47 ± 0.64	64.16 ± 0.92	**85.32 ± 0.60**	**85.45 ± 0.86**	71.26	71.26
20% + $\mathcal{L}_{gm}$	**87.2 ± 0.99**	**86.87 ± 0.78**	**49.68 ± 0.55**	**50.40 ± 0.46**	**63.58 ± 0.08**	**64.21 ± 0.78**	85.25 ± 0.90	85.52 ± 0.24	**71.42**	**71.75**
Gaussian	85.22 ± 0.24	86.82 ± 0.72	46.15 ± 0.51	49.59 ± 0.76	63.72 ± 0.35	**64.40 ± 0.76**	84.06 ± 0.74	**83.53 ± 0.70**	69.78	71.08
Gaussian + $\mathcal{L}_{gm}$	**85.94 ± 0.55**	**87.25 ± 0.36**	**48.23 ± 0.69**	**50.29 ± 0.70**	**64.06 ± 0.73**	64.29 ± 0.94	**84.46 ± 0.33**	83.29 ± 0.47	**70.67**	**71.45**

Table 1: Accuracy on 4 text classification benchmark. 0% represents a model trained on P^c, 1% and so on denote the proportion of random noise, and Gaussian refers to Gaussian noise. * cited from Saunshi et al. (2021).

	BBC		Balanced COPA		MRPC		WiC		Avg	
	Linear	MLP	Linear	MLP	Linear	MLP	Linear	MLP	Linear	MLP
Llama-3-8B	96.90 ± 0.40	97.50 ± 0.20	69.00 ± 0.20	**65.60 ± 0.50**	72.00 ± 0.81	67.53 ± 0.93	64.14 ± 0.56	59.07 ± 0.34	75.51	72.42
Llama-3-8B + $\mathcal{L}_{gm}$	**98.00 ± 0.50**	**98.20 ± 0.40**	**70.80 ± 1.70**	64.80 ± 0.20	**74.89 ± 0.40**	**74.14 ± 1.49**	**64.71 ± 0.94**	**64.21 ± 0.83**	**77.10**	**75.33**
Llama-3-8B-Instruct	96.80 ± 0.70	96.90 ± 0.30	87.80 ± 0.70	88.80 ± 0.60	72.57 ± 0.26	71.42 ± 0.13	65.92 ± 0.53	61.85 ± 0.59	80.77	79.74
Llama-3-8B-Instruct + $\mathcal{L}_{gm}$	**97.70 ± 0.20**	**97.80 ± 0.40**	**88.40 ± 0.90**	**89.60 ± 0.50**	**77.79 ± 0.58**	**76.81 ± 0.20**	**68.64 ± 0.26**	**67.71 ± 0.51**	**83.13**	**82.98**
Llama-3.2-3B-Instruct	97.30 ± 0.60	97.20 ± 0.80	80.40 ± 0.90	**79.60 ± 0.20**	77.79 ± 0.52	72.57 ± 0.31	64.07 ± 0.82	57.50 ± 0.35	79.89	76.71
Llama-3.2-3B-Instruct + $\mathcal{L}_{gm}$	**97.60 ± 0.10**	**97.80 ± 0.30**	**81.60 ± 1.00**	79.40 ± 0.10	**78.43 ± 0.78**	**76.57 ± 1.12**	**64.35 ± 0.62**	**62.64 ± 0.07**	**80.49**	**79.10**
Qwen2.5-1.5B-Instruct	97.00 ± 0.30	96.60 ± 0.70	80.80 ± 0.70	82.20 ± 0.50	74.49 ± 0.71	73.39 ± 0.90	65.92 ± 0.45	61.64 ± 0.20	79.55	78.45
Qwen2.5-1.5B-Instruct + $\mathcal{L}_{gm}$	**97.40 ± 0.10**	**97.20 ± 0.80**	**84.00 ± 0.90**	**83.40 ± 0.30**	**79.65 ± 0.62**	**78.37 ± 0.84**	**67.71 ± 0.49**	**66.92 ± 0.55**	**82.19**	**81.47**
Qwen2.5-7B-Instruct	96.30 ± 0.30	96.70 ± 0.50	94.60 ± 0.90	95.80 ± 0.40	83.71 ± 0.92	76.81 ± 0.51	68.92 ± 0.41	64.92 ± 0.18	85.88	83.55
Qwen2.5-7B-Instruct + $\mathcal{L}_{gm}$	**97.10 ± 0.80**	**97.40 ± 0.20**	**95.60 ± 0.50**	**96.00 ± 0.80**	**84.98 ± 0.12**	**83.13 ± 0.49**	**72.28 ± 0.98**	**70.14 ± 0.94**	**87.49**	**86.66**

Table 2: Accuracy of LLMs on 4 natural language understanding benchmark.

work in visual domains (Zada et al., 2022), suggesting that incorporating random noise into training sets might enhance model robustness. Additionally, we observe that the performance of models subjected to Gaussian noise varies across different datasets. These observations warrant further investigation.

3.5 Broader Impact of the Results

In addition to providing explanations regarding the impact of random noise on pre-training language models, we aim to extend our proposed theory to other areas, therefore demonstrating the practical value of our research findings.

One immediate direction is the training of multilingual models (Pires et al., 2019; Chi et al., 2020; Yang et al., 2024a). Clearly, tokens corresponding to different languages are distinct, and their distributions naturally satisfy Assumption 1. For example, in an English-French bilingual model, let P^c represent English and P^n represent French. Supposing the pre-training corpus consists of an equal distribution of English and French, and given that the two distributions are similar, we can assume that $p_c \approx p_n$, leading to $\epsilon \approx 0$. This provides a theoretical foundation for the success of multilingual models. See Appendix D.3 for more details.

Beyond language modality, random white noise has received increased attention in the speech domain (Chen et al., 2022, 2021; Yang et al., 2024c, 2023b). Since our theory applies to any cross-entropy-like loss, it can also explain why speech models pre-trained on very noisy large-scale datasets, such as Gigaspeech (Chen et al., 2021), which contain significant background noise and prolonged silence at the beginning and end of a few samples, still perform remarkably well.

4 Reducing the Noise with Local Gradient Matching

In Section 3, we know that the influence of noise on NTP loss is rather small. However, Figure 3(c) and Table 1 show that the Gaussian noise-trained model with lower NTP loss suffers a 1.5% decrease in accuracy in downstream tasks. Although this might be mitigated during the pre-training phase, considering that most practitioners fine-tune pre-trained models rather than training them from scratch, we propose a novel black-box fine-tuning method termed Local Gradient Matching loss to tame the influence. Extensive experiments across 8 natural language understanding and 14 image classification benchmark datasets further demonstrate that the proposed method consistently enhances performance across different backbones and modalities. We also provide a theoretical analysis.

4.1 Method

In the preceding analysis, we demonstrate that the population-level loss function is only marginally affected by random noise. However, during the SGD training process, its presence introduces cer-

Model	EfficientNet-B3		ResNetv2-152x2		Swin-L		ConvNext-L		ViT-L	
Pre-training Data	JFT-300M		ImageNet-21K		ImageNet-21K		Laion-2B		Laion-2B	
Fine-tuning Method	Linear	MLP	Linear	MLP	Linear	MLP	Linear	MLP	Linear	MLP
w/o $\mathcal{L}_{gm}$	73.27	**76.62**	78.14	79.60	81.43	84.19	82.89	85.71	86.86	89.12
w/ $\mathcal{L}_{gm}$	**74.02**	75.90	**79.49**	**79.94**	**82.70**	**84.42**	**84.07**	**86.27**	**88.03**	**89.31**

Table 3: Average accuracy of 5 vision backbone models on **14** commonly-used vision datasets.

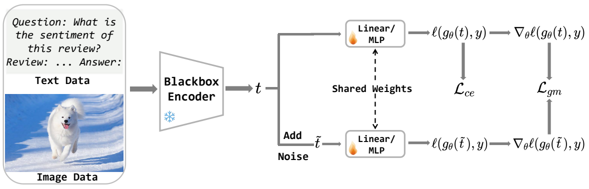

Question: What is the sentiment of this review? Review: ... Answer:
Text Data
Image Data
Blackbox Encoder
t
Add Noise
$\tilde{t}$
Linear/ MLP
Linear/ MLP
Shared Weights
$\ell(g_\theta(t), y)$
$\nabla_\theta \ell(g_\theta(t), y)$
$\ell(g_\theta(\tilde{t}), y)$
$\nabla_\theta \ell(g_\theta(\tilde{t}), y)$
$\mathcal{L}_{ce}$
$\mathcal{L}_{gm}$

Figure 5: Overview of the proposed Local Gradient Mathcing scheme.

	RTE	MRPC	CoLA	STS-B
L^2-SP*	70.58	**87.74**	60.54	89.38
L^2-SP + $\mathcal{L}_{gm}$	**71.25**	87.62	**61.79**	**89.62**
SMART*	72.23	87.86	63.16	90.11
SMART + $\mathcal{L}_{gm}$	**72.94**	**88.61**	**63.28**	**90.42**
LNSR*	73.31	88.50	63.35	90.23
LNSR + $\mathcal{L}_{gm}$	**73.95**	**89.42**	**63.82**	**90.47**

Table 4: Evaluation of our method combined with SOTA fine-tuning techniques utilizing BERT-Large as the backbone model across 4 datasets. * cited from Hua et al. (2021)

tain noise into the gradients. Prior studies (Chen et al., 2023; Xie et al., 2021c) have shown that artificially added gradient noise can hurt the model's generalization. Therefore, inspired by Sharpness-Aware Minimization (SAM) (Foret et al., 2021; Zhang et al., 2023; Zhao et al., 2022; Wen et al., 2023) and noise-robust fine-tuning methods (Hua et al., 2023, 2021; Jiang et al., 2020), we propose explicitly enhancing the denoising capabilities of the downstream task head by aligning local gradients.

Specifically, let C denote the number of classes in the downstream task, and let $g_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^C$ represent the linear or MLP classification head parameterized by θ. Let t^* be the feature extracted by h^*, t be the feature extracted by h, and y be the corresponding label. Let $\ell(\hat{y}, y)$ be the loss function(typically cross-entropy), and $\mathcal{L}_{ce}(\mathcal{D}, g_\theta) = \mathbb{E}_{(t,y)\sim\mathcal{D}}\ell(g_\theta(t), y)$ be the population-level loss where $\mathcal{D}$ represents the joint distribution of downstream features and labels. Due to the additional randomness introduced by h as a result of noise, t can be viewed as t^* perturbed by minor disturbances. If both t^* and t were known, their distribution could be aligned to achieve denoising. However, in practical applications, it is challenging to obtain t^*. To construct contrastive sample pairs without t^*, we add Gaussian noise to t to obtain $\hat{t}$:

$$\hat{t} = t + \gamma \cdot \delta \tag{3}$$

where $\delta \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$ denotes the standard normal distribution noise. Our objective is to minimize the discrepancy between the distributions of $g_\theta(t)$ and $g_\theta(\hat{t})$. Instead of the conventional regularization term $||g_\theta(t) - g_\theta(\hat{t})||_2$, we propose to align the gradient difference:

$$\begin{aligned}\mathcal{L}_{gm}(\theta) = ||&\mathbb{E}_{(t,y)\sim\mathcal{D}}\nabla_\theta \ell(g_\theta(t), y) \\ &- \mathbb{E}_{(\hat{t},y)\sim\hat{\mathcal{D}}}\nabla_\theta \ell(g_\theta(\hat{t}), y)||_2\end{aligned} \tag{4}$$

Intuitively, if the gradients with respect to t and $\hat{t}$ can be perfectly aligned, then the classification head is insensitive to small perturbations in the input, suggesting that it possesses some denoising capability. Consequently, it should be able to mitigate the noise in t, bringing it closer to t^*.

4.2 Theoretical Analysis

To theoretically support the proposed method, we investigate the properties of Equation (4) and find that it can be upper bounded by the smoothness, input flatness, and loss function value at θ. Concretely, since we set γ in Equation (3) to be small, the perturbation can be considered to distribute within an open ball $B(0, \rho)$. Consequently, we have the following result:

Proposition 2. *Suppose $\ell(g_\theta(t), y)$ is β-smooth with ρ-input flatness $R_\rho(\theta)$ (c.f. Appendix B.4), for any $\theta \in \Theta$:*

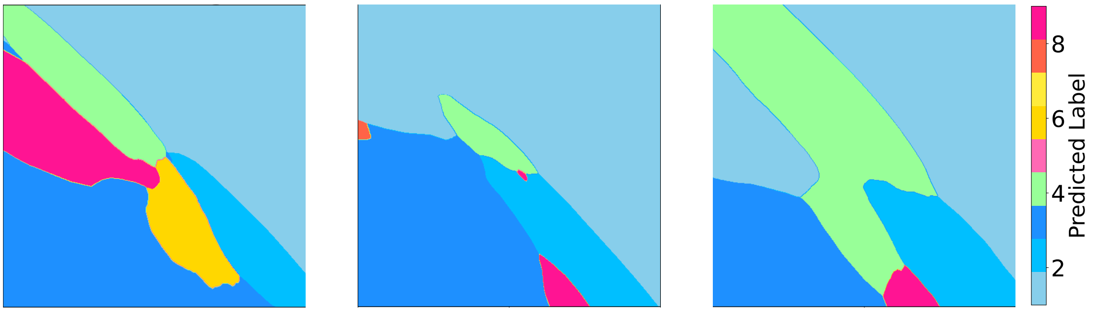
$$\mathcal{L}_{gm}(\theta) \leq 2\beta + 2\mathcal{L}_{ce}(\mathcal{D}, g_\theta) + R_\rho(\theta). \tag{5}$$

Proposition 2 demonstrates that $\mathcal{L}_{gm}$ is closely associated with the smoothness of the loss function in both the parameter space and the input space. As a flat minima is widely acknowledged to benefit the generalization of neural networks (Xie et al., 2021b; Baldassi et al., 2021), it explains the effectiveness of $\mathcal{L}_{gm}$. The final loss function is:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda\mathcal{L}_{gm} \tag{6}$$

4.3 Experiments

We first conduct extensive experiments using trained GPT-2 models. Then, to further validate the novelty and effectiveness of the LGM loss, we conduct additional experiments using Llama-3 (Dubey



(a) No regularization. (b) With L^2 regularization. (c) With $\mathcal{L}_{gm}$.

Figure 6: Visualization of input sensitivity for models trained with (a) no (b) L^2 (c) $\mathcal{L}_{gm}$ regularization. We randomly select a sample and introduce perturbations on a two-dimensional hyperplane, where different colors represent different labels, and green indicates the correct label.

Method	SST-2	
	Linear	MLP
0%	86.71	87.36
0% + $\|\|\nabla_\theta \ell(g_\theta(t), y)\|\|_2$	87.04	87.24
0% + $\cos(\nabla_\theta \ell(g_\theta(t), y), \nabla_\theta \ell(g_\theta(\hat{t}), y))$	86.89	87.52
0% + $\mathcal{L}_{gm}$	**87.42**	**87.86**

Table 5: Ablation Study. To investigate the effects of reducing $\mathcal{L}_{gm}$, experiments are conducted to examine the impact of separately reducing the norm versus increasing the cosine similarity.

γ	λ	DTD	
		Linear	MLP
0.001	0.001	76.31	78.54
0.05	0.05	76.54	79.51
0.1	0.1	76.43	79.12

Table 6: Hyperparameter sensitivity experiments on DTD with ConvNext as the backbone.

et al., 2024) and vision models. These experiments are intended to showcase the generalizability of our approach beyond the specific context of GPT-2, demonstrating its applicability across different types of models and tasks. While these experiments enrich our study, **they are not the core focus but rather supplementary evidence** supporting the broader applicability of our proposed solution. Details can be found in Appendix E.

We validate the performance of $\mathcal{L}_{gm}$ on models pre-trained with noisy data using four commonly used classification datasets: SST-2, SST-fine, 20newsgroup, and CR. The training hyperparameters follow those of Saunshi et al. (2021), where $\gamma = 0.01$ and $\lambda = 0.15$ apply to all four experiments. In line with the approach described by Chen et al. (2024), we freeze the model parameters and only fine-tune a linear or MLP classifier head. As shown in Table 1, our model achieves competitive results without reaching the number of training iterations of GPT-2, and $\mathcal{L}_{gm}$ consistently boosts performance.

Results in Table 2 indicate that our method provides a 3% improvement across multiple NLU datasets with LLM backbone. In addition, we select five commonly used backbone models in the visual domain and conduct experiments on fourteen datasets. The results are shown in Table 3. It can be seen that our method is equally applicable to visual tasks, achieving a performance improvement of more than 1% under the linear probe setting.

Furthermore, we visualize the sensitivity of different regularization terms to input perturbations, as illustrated in Figure 6. Compared with other regularization methods, our loss function can increase the size of the region for correct decisions, thereby enhancing the model's robustness to input perturbations. We also carry out ablation studies and parameter sensitivity analyses, with results presented in Table 5 and Table 6, which all demonstrate the effectiveness and robustness of LGM.

5 Conclusion

In this paper, we investigate the random noise present in language model pre-training datasets, which is inevitable in real-world scenarios but receives little attention. We pre-train multiple GPT-2 models under varying noise levels and find that random noise has a minor impact on the pre-training loss. We then provide a theoretical explanation for this phenomenon and discover that our theory can elucidate the success of multilingual models. Interestingly, we observe that slight noise can sometimes enhance a model's generalization ability. Then, building on the noisy model learning setup, we propose a novel local gradient matching loss. Extensive experiments across multiple datasets in both language and vision tasks, as well as with various backbone models, validate the effectiveness of our proposed method. We hope this work inspires more researchers to focus on data-centric AI.

Limitations

In this section, we discuss the limitations of this paper.

Firstly, due to limitations in computational resources and costs, we pre-train only the GPT-2 124M and 774M model(see Appendix D.5) on the OpenWebText dataset. Compared to today's large language models, both the scale of OpenWebText and that of GPT-2 are relatively small. Additionally, the types of noise considered are limited to uniform and Gaussian distributions. However, based on Proposition 1, we argue that training GPT-2 on the Synthetic OpenWebText dataset is sufficient to uncover the essence of the issue, as Proposition 1 makes no assumptions about data distribution or model architecture.

Secondly, on the theoretical front, we consider neural networks as black boxes and focus on analyzing the properties of global minima. Due to limited mathematical skills, we do not delve into the dynamical aspects to specifically examine how random noise within datasets influences model gradients, nor do we explore the differences between global and local minima obtained through stochastic gradient descent. However, experimental results indicate that neural networks trained with stochastic gradient descent do not suffer from significant disturbances.

References

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. 2024. A survey on data selection for language models. *Transactions on Machine Learning Research*. Survey Certification.

Zeyuan Allen-Zhu and Yuanzhi Li. 2024a. Physics of language models: Part 3.1, knowledge storage and extraction. In *Forty-first International Conference on Machine Learning*.

Zeyuan Allen-Zhu and Yuanzhi Li. 2024b. Physics of language models: Part 3.3, knowledge capacity scaling laws. *arXiv preprint arXiv:2404.05405*.

Carlo Baldassi, Clarissa Lauditi, Enrico M. Malatesta, Gabriele Perugini, and Riccardo Zecchina. 2021. Unveiling the structure of wide flat minima in neural networks. *Phys. Rev. Lett.*, 127:278301.

David Barrett and Benoit Dherin. 2021. Implicit gradient regularization. In *International Conference on Learning Representations*.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning*, 79:151–175.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Yekun Chai, Qingyi Liu, Shuohuan Wang, Yu Sun, Qiwei Peng, and Hua Wu. 2024. On training data influence of GPT models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3126–3150, Miami, Florida, USA. Association for Computational Linguistics.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar. Association for Computational Linguistics.

Feng Chen, Daniel Kunin, Atsushi Yamamura, and Surya Ganguli. 2023. Stochastic collapse: How gradient noise attracts SGD dynamics towards simpler subnetworks. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, et al. 2021. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. In *22nd Annual Conference of the International Speech Communication Association, INTERSPEECH 2021*, pages 4376–4380. International Speech Communication Association.

Hao Chen, Jindong Wang, Ankit Shah, Ran Tao, Hongxin Wei, Xing Xie, Masashi Sugiyama, and Bhiksha Raj. 2024. Understanding and mitigating the label noise in pre-training on downstream tasks. In *The Twelfth International Conference on Learning Representations*.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. 2017. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883.

Valeriia Cherepanova and James Zou. 2024. Talking nonsense: Probing large language models' understanding of adversarial gibberish inputs. *arXiv preprint arXiv:2404.17120*.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. 2014. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613.

Sayantan Dasgupta, Trevor Cohn, and Timothy Baldwin. 2023. Cost-effective distillation of large language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7346–7354, Toronto, Canada. Association for Computational Linguistics.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third international workshop on paraphrasing (IWP2005)*.

Alexey Dosovitskiy. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Zhekai Du and Jingjing Li. 2023. Diffusion-based probabilistic uncertainty estimation for active domain adaptation. In *Advances in Neural Information Processing Systems*, volume 36, pages 17129–17155. Curran Associates, Inc.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yanai Elazar, Akshita Bhagia, Ian Helgi Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Evan Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, Hannaneh Hajishirzi, Noah A. Smith, and Jesse Dodge. 2024. What's in my big data? In *The Twelfth International Conference on Learning Representations*.

Qihang Fan, Huaibo Huang, Mingrui Chen, and Ran He. 2024a. Semantic equitable clustering: A simple, fast and effective strategy for vision transformer. *arXiv preprint arXiv:2405.13337*.

Qihang Fan, Huaibo Huang, Mingrui Chen, and Ran He. 2024b. Vision transformer with sparse scan prior. *arXiv preprint arXiv:2405.13335*.

Qihang Fan, Huaibo Huang, Mingrui Chen, Hongmin Liu, and Ran He. 2024c. Rmt: Retentive networks meet vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5641–5651.

Zhen Fang, Yixuan Li, Jie Lu, Jiahua Dong, Bo Han, and Feng Liu. 2022. Is out-of-distribution detection learnable? In *Advances in Neural Information Processing Systems*, volume 35, pages 37199–37213. Curran Associates, Inc.

Li Fei-Fei, Rob Fergus, and Pietro Perona. 2004a. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE.

Li Fei-Fei, Rob Fergus, and Pietro Perona. 2004b. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 2021. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Yang Gao, Zunlei Feng, Xiaoyang Wang, Mingli Song, Xingen Wang, Xinyu Wang, and Chun Chen. 2023. Reinforcement learning based web crawler detection for diversity and dynamics. *Neurocomputing*, 520:115–128.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2019. Openwebtext corpus.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2019. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226.

Geoffrey Hinton. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Hang Hua, Xingjian Li, Dejing Dou, Cheng-Zhong Xu, and Jiebo Luo. 2023. Improving pretrained language model fine-tuning with noise stability regularization. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15.

Hang Hua, Xingjian Li, Dejing Dou, Chengzhong Xu, and Jiebo Luo. 2021. Noise stability regularization for improving BERT fine-tuning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3229–3241, Online. Association for Computational Linguistics.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, Online. Association for Computational Linguistics.

Renren Jin, Jiangcun Du, Wuwei Huang, Wei Liu, Jian Luan, Bin Wang, and Deyi Xiong. 2024. A comprehensive evaluation of quantization strategies for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12186–12215, Bangkok, Thailand. Association for Computational Linguistics.

Wooyoung Kang, Jonghwan Mun, Sungjun Lee, and Byungseok Roh. 2023. Noise-aware learning from web-crawled image-text data for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2942–2952.

Pride Kavumba, Naoya Inoue, Benjamin Heinzerling, Keshav Singh, Paul Reisert, and Kentarou Inui. 2020. Balanced copa: Countering superficial cues in causal reasoning. *Association for Natural Language Processing*, pages 1105–1108.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561.

Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images.

Eldar Kurtic, Daniel Campos, Tuan Nguyen, Elias Frantar, Mark Kurtz, Benjamin Fineran, Michael Goin, and Dan Alistarh. 2022. The optimal BERT surgeon: Scalable and accurate second-order pruning for large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4163–4181, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jean-François Le Gall. 2022. *Measure theory, probability, and stochastic processes*. Springer.

Alycia Lee, Brando Miranda, Sudharsan Sundar, and Sanmi Koyejo. 2023. Beyond scale: the diversity coefficient as a data quality metric demonstrates llms are pre-trained on formally diverse data. *arXiv preprint arXiv:2306.13840*.

Jingjing Li, Zhekai Du, Lei Zhu, Zhengming Ding, Ke Lu, and Heng Tao Shen. 2021. Divergence-agnostic unsupervised domain adaptation by adversarial attacks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8196–8211.

Jingjing Li, Zhiqi Yu, Zhekai Du, Lei Zhu, and Heng Tao Shen. 2024a. A comprehensive survey on source-free domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5743–5762.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2024b. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7602–7635, Mexico City, Mexico. Association for Computational Linguistics.

Hong Liu, Sang Michael Xie, Zhiyuan Li, and Tengyu Ma. 2023. Same pre-training loss, better downstream: Implicit bias matters for language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22188–22214. PMLR.

Ruikang Liu, Haoli Bai, Haokun Lin, Yuening Li, Han Gao, Zhengzhuo Xu, Lu Hou, Jun Yao, and Chun Yuan. 2024. IntactKV: Improving large language model quantization by keeping pivot tokens intact. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7716–7741, Bangkok, Thailand. Association for Computational Linguistics.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2024. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3245–3276, Mexico City, Mexico. Association for Computational Linguistics.

I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. 2020. Does label smoothing mitigate label noise? In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6448–6458. PMLR.

Ailong Ma, Chenyu Zheng, Junjue Wang, and Yanfei Zhong. 2023. Domain adaptive land-cover classification via local consistency and global diversity. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–17.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. 2013. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*.

Lichao Meng, Hongzu Su, Chunwei Lou, and Jingjing Li. 2022. Cross-domain mutual information adversarial maximization. *Engineering Applications of Artificial Intelligence*, 110:104665.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 4. Granada.

Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yu Pan, Ye Yuan, Yichun Yin, Jiaxin Shi, Zenglin Xu, Ming Zhang, Lifeng Shang, Xin Jiang, and Qun Liu. 2024. Preparing lessons for progressive training on language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18860–18868.

Yu Pan, Ye Yuan, Yichun Yin, Zenglin Xu, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Reusing pretrained models by multi-linear operators for efficient training. *Advances in Neural Information Processing Systems*, 36:3248–3262.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. 2012. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. 2021. Imagenet-21k pretraining for the masses. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Jinghan Ru, Jun Tian, Chengwei Xiao, Jingjing Li, and Heng Tao Shen. 2024. Imbalanced open set domain adaptation via moving-threshold estimation and gradual alignment. *IEEE Transactions on Multimedia*, 26:2504–2514.

Antony Samuels and John Mcgonical. 2020. News sentiment analysis. *arXiv preprint arXiv:2007.02238*.

Nikunj Saunshi, Sadhika Malladi, and Sanjeev Arora. 2021. A mathematical exploration of why language models help solve downstream tasks. In *International Conference on Learning Representations*.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems*, volume 35, pages 25278–25294. Curran Associates, Inc.

Mohamed El Amine Seddik, Suei-Wen Chen, Soufiane Hayou, Pierre Youssef, and Merouane Abdelkader DEBBAH. 2024. How bad is training on synthetic data? a statistical analysis of language model collapse. In *First Conference on Language Modeling*.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Anastasia Kozlova, Vladislav Mikhailov, and Tatiana Shavrina. 2024. mGPT: Few-shot learners go multilingual. *Transactions of the Association for Computational Linguistics*, 12:58–79.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. 2024. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE transactions on neural networks and learning systems*, 34(11):8135–8153.

Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. 2010. Smoothness, low noise and fast rates. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852.

I Sutskever. 2014. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*.

Mingxing Tan and Quoc Le. 2019. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR.

Linjie Tang, Pengfei Yi, Mingrui Chen, MingKun Yang, and Dingkang Liang. 2024. Not all texts are the same: Dynamically querying texts for scene text detection. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 363–377. Springer.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. 2018. Rotation equivariant cnns for digital pathology. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*, pages 210–218. Springer.

Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2020. Structured pruning of large language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6151–6162, Online. Association for Computational Linguistics.

Colin Wei, Sang Michael Xie, and Tengyu Ma. 2021. Why do pretrained language models help in downstream tasks? an analysis of head and prompt tuning. In *Advances in Neural Information Processing Systems*, volume 34, pages 16158–16170. Curran Associates, Inc.

Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. 2023. How sharpness-aware minimization minimizes sharpness? In *The Eleventh International Conference on Learning Representations*.

Zhiyuan Weng, Guikun Chen, and Wenguan Wang. 2025. Do as we do, not as you think: the conformity of large language models. *arXiv preprint arXiv:2501.13381*.

Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. 2023. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16133–16142.

Kangkai Wu, Jingjing Li, Lichao Meng, Fengling Li, and Ke Lu. 2024. Online adaptive fault diagnosis with test-time domain adaptation. *IEEE Transactions on Industrial Informatics*, pages 1–11.

Mingyu Xiao, Jianan Zhang, and Weibo Lin. 2022. Parameterized algorithms and complexity for the traveling purchaser problem and its variants. *Journal of Combinatorial Optimization*, pages 1–17.

Sang Michael Xie, Tengyu Ma, and Percy Liang. 2021a. Composed fine-tuning: Freezing pre-trained denoising autoencoders for improved generalization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11424–11435. PMLR.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2023a. Doremi: Optimizing data mixtures speeds up language model pretraining. In *Advances in Neural Information Processing Systems*, volume 36, pages 69798–69818. Curran Associates, Inc.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. 2023b. Data selection for language models via importance resampling. In *Advances in Neural Information Processing Systems*, volume 36, pages 34201–34227. Curran Associates, Inc.

Yuxin Xie, Zhihong Zhu, Xianwei Zhuang, Liming Liang, Zhichang Wang, and Yuexian Zou. 2024. Gpa: global and prototype alignment for audio-text retrieval. In *Proc. Interspeech 2024*, pages 5078–5082.

Zeke Xie, Issei Sato, and Masashi Sugiyama. 2021b. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *International Conference on Learning Representations*.

Zeke Xie, Xinrui Wang, Huishuai Zhang, Issei Sato, and Masashi Sugiyama. 2022. Adaptive inertia: Disentangling the effects of adaptive learning rate and momentum. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 24430–24459. PMLR.

Zeke Xie, Zhiqiang Xu, Jingzhao Zhang, Issei Sato, and Masashi Sugiyama. 2023c. On the overlooked pitfalls of weight decay and how to mitigate them: A gradient-norm perspective. In *Advances in Neural Information Processing Systems*, volume 36, pages 1208–1228. Curran Associates, Inc.

Zeke Xie, Li Yuan, Zhanxing Zhu, and Masashi Sugiyama. 2021c. Positive-negative momentum: Manipulating stochastic gradient noise to improve generalization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11448–11458. PMLR.

Bang Yang, Fenglin Liu, Yuexian Zou, Xian Wu, Yaowei Wang, and David A. Clifton. 2024a. Zeronlg: Aligning and autoencoding domains for zero-shot multimodal and multilingual natural language generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5712–5724.

Dongchao Yang, Haohan Guo, Yuanyuan Wang, Rongjie Huang, Xiang Li, Xu Tan, Xixin Wu, and Helen Meng. 2024b. Uniaudio 1.5: Large language model-driven audio codec is a few-shot audio task learner. *arXiv preprint arXiv:2406.10056*.

Dongchao Yang, Songxiang Liu, Rongjie Huang, Jinchuan Tian, Chao Weng, and Yuexian Zou. 2023a. Hifi-codec: Group-residual vector quantization for high fidelity audio codec. *arXiv preprint arXiv:2305.02765*.

Dongchao Yang, Songxiang Liu, Rongjie Huang, Chao Weng, and Helen Meng. 2024c. Instructtts: Modelling expressive tts in discrete latent space with natural language style prompt. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:2913–2925.

Dongchao Yang, Jinchuan Tian, Xu Tan, Rongjie Huang, Songxiang Liu, Xuankai Chang, Jiatong Shi, Sheng Zhao, Jiang Bian, Xixin Wu, et al. 2023b. Uniaudio: An audio foundation model toward universal audio generation. *arXiv preprint arXiv:2310.00704*.

Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. 2023c. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1720–1733.

Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. 2023d. Dataset pruning: Reducing training data by examining generalization influence. In *The Eleventh International Conference on Learning Representations*.

Wenyuan Yang, Yuguo Yin, Gongxi Zhu, Hanlin Gu, Lixin Fan, Xiaochun Cao, and Qiang Yang. 2023e. Fedzkp: Federated model ownership verification with zero-knowledge proof. *arXiv preprint arXiv:2305.04507*.

Wenyuan Yang, Gongxi Zhu, Yuguo Yin, Hanlin Gu, Lixin Fan, Qiang Yang, and Xiaochun Cao. 2023f. Fedsov: Federated model secure ownership verification with unforgeable signature. *arXiv preprint arXiv:2305.06085*.

Yuqi Ye and Wei Gao. 2024. Llm-pcgc: Large language model-based point cloud geometry compression. *arXiv preprint arXiv:2408.08682*.

Shiran Zada, Itay Benou, and Michal Irani. 2022. Pure noise to the rescue of insufficient data: Improving imbalanced classification by training on random noise images. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 25817–25833. PMLR.

Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. 2021. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*.

Xingxuan Zhang, Renzhe Xu, Han Yu, Hao Zou, and Peng Cui. 2023. Gradient norm aware minimization seeks first-order flatness and improves generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20247–20257.

Xuchao Zhang, Fanglan Chen, Chang-Tien Lu, and Naren Ramakrishnan. 2019. Mitigating uncertainty in document classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3126–3136, Minneapolis, Minnesota. Association for Computational Linguistics.

Yingtao Zhang, Haoli Bai, Haokun Lin, Jialin Zhao, Lu Hou, and Carlo Vittorio Cannistraci. 2024. Plug-and-play: An efficient post-training pruning method for large language models. In *The Twelfth International Conference on Learning Representations*.

Jiaqi Zhao, Miao Zhang, Chao Zeng, Ming Wang, Xuebo Liu, and Liqiang Nie. 2024. LRQuant: Learnable and robust post-training quantization for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2240–2255, Bangkok, Thailand. Association for Computational Linguistics.

Yang Zhao, Hao Zhang, and Xiuyuan Hu. 2022. Penalizing gradient norm for efficiently improving generalization in deep learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 26982–26992. PMLR.

Chenyu Zheng, Guoqiang Wu, Fan Bao, Yue Cao, Chongxuan Li, and Jun Zhu. 2023a. Revisiting discriminative vs. generative classifiers: Theory and implications. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 42420–42477. PMLR.

Chenyu Zheng, Guoqiang Wu, and Chongxuan LI. 2023b. Toward understanding generative data augmentation. In *Advances in Neural Information Processing Systems*, volume 36, pages 54046–54060. Curran Associates, Inc.

Jing Zhou, Chenglin Jiang, Wei Shen, Xiao Zhou, and Xiaonan He. 2024. Leveraging web-crawled data for high-quality fine-tuning. *arXiv preprint arXiv:2408.08003*.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022a. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16816–16825.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022b. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348.

Xianwei Zhuang, Xuxin Cheng, Liming Liang, Yuxin Xie, Zhichang Wang, Zhiqi Huang, and Yuexian Zou. 2024a. Pcad: Towards asr-robust spoken language understanding via prototype calibration and asymmetric decoupling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5235–5246.

Xianwei Zhuang, Xuxin Cheng, Zhihong Zhu, Zhanpeng Chen, Hongxiang Li, and Yuexian Zou. 2024b. Towards multimodal-augmented pre-trained language models via self-balanced expectation-maximization iteration. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 4670–4679.

Xianwei Zhuang, Xuxin Cheng, and Yuexian Zou. 2024c. Towards explainable joint models via information theory for multiple intent detection and slot filling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19786–19794.

Xianwei Zhuang, Hongxiang Li, Xuxin Cheng, Zhihong Zhu, Yuxin Xie, and Yuexian Zou. 2024d. Kdpror: A knowledge-decoupling probabilistic framework for video-text retrieval. In *European Conference on Computer Vision*, pages 313–331. Springer.

Xianwei Zhuang, Hualiang Wang, Xiaoxuan He, Siming Fu, and Haoji Hu. 2025a. Semigmmpoint: Semi-supervised point cloud segmentation based on gaussian mixture models. *Pattern Recognition*, 158:111045.

Xianwei Zhuang, Zhichang Wang, Xuxin Cheng, Yuxin Xie, Liming Liang, and Yuexian Zou. 2024e. Macsc: Towards multimodal-augmented pre-trained language models via conceptual prototypes and self-balancing calibration. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8070–8083.

Xianwei Zhuang, Yuxin Xie, Yufan Deng, Liming Liang, Jinghan Ru, Yuguo Yin, and Yuexian Zou. 2025b. Vargpt: Unified understanding and generation in a visual autoregressive multimodal large language model. *arXiv preprint arXiv:2501.12327*.

Xianwei Zhuang, Zhihong Zhu, Zhanpeng Chen, Yuxin Xie, Liming Liang, and Yuexian Zou. 2024f. Game on tree: Visual hallucination mitigation via coarse-to-fine view tree and game theory. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17984–18003.

Xianwei Zhuang, Zhihong Zhu, Yuxin Xie, Liming Liang, and Yuexian Zou. 2025c. Vasparse: Towards efficient visual hallucination mitigation for large vision-language model via visual-aware sparsification. *arXiv preprint arXiv:2501.06553*.

A Notations

The commonly used notations and their descriptions are as follows.

Notation	Description
L	context length
d	embedding dimension
$\mathcal{W}$	vocabulary of words
$V = \|\mathcal{W}\|$	vocabulary size
$\mathcal{X} = \cup_{i=1}^{L} \mathcal{W}^i$	model input space
$\mathcal{H}$	model space
$h : \mathcal{X} \to \mathbb{R}^V \in \mathcal{H}$	language model
Δ_A	distribution defined on a discrete set A
$P^c \in \Delta_{\mathcal{X}\times\mathcal{W}}$	distribution of clean data
$P^n \in \Delta_{\mathcal{X}\times\mathcal{W}}$	distribution of pure noise data
$P^m \in \Delta_{\mathcal{X}\times\mathcal{W}}$	distribution of mixed noisy data
α	proportion of noise in training data
P_X	marginal distribution of the joint distribution P
$P_{\cdot\|X}$	conditional distribution of the joint distribution P
$\boldsymbol{p}^h_{\cdot\|x}(w)$	the w-th dimension of the probability distribution corresponding to $h(x)$
$\mathrm{supp}(P^c)$	support of distribution P^c
$\mathcal{L}_{ntp}(P, h)$	next-token prediction loss of model h on the distribution P
$g_\theta : \mathbb{R}^d \to \mathbb{R}^C$	downstream classification head
$\theta \in \Theta$	parameters of g
$t \in \mathcal{T}$	feature of downstream task data extracted by backbone model
$y \in \mathcal{Y}$	label of downstream task data
$C = \|\mathcal{Y}\|$	number of classes of the downstream task
$\ell(\hat{y}, y)$	downstream task loss function, typically cross-entropy
$\mathcal{D}$	joint distribution of downstream feature and label
$\mathcal{L}_{ce}(\mathcal{D}, g_\theta)$	population-level loss with downstream data distribution $\mathcal{D}$ and head g_θ

Table 7: Nomenclature.

B Proofs

B.1 Explanation of Equation (1)

Let $\mathcal{M}$ be a measurable space, and let P_1 and P_2 be probability measures defined on this space. We assume that N_1 samples are drawn from P_1 and N_2 samples from P_2. Define $\mu = \frac{N_1}{N_1+N_2}$, so that $1 - \mu = \frac{N_2}{N_1+N_2}$.

We aim to show that this collection of $N_1 + N_2$ samples can be regarded as drawn from a mixed distribution

$$P_3 = \mu P_1 + (1 - \mu) P_2$$

First, define a new probability measure P_3 as $P_3(A) = \alpha P_1(A) + (1 - \alpha) P_2(A)$ for any measurable set $A \subseteq \mathcal{M}$. Here, P_3 is a convex combination of P_1 and P_2, and thus P_3 is also a valid probability measure (Le Gall, 2022).

For any measurable set $A \subseteq \mathcal{M}$, we examine the probability that a single sample point falls in A by law of total probability:

- A sample from P_1 is selected with probability μ, and within this case, the probability of landing in A is $P_1(A)$.

- A sample from P_2 is selected with probability $1 - \mu$, and the probability of it falling in A is $P_2(A)$.

Thus, the probability of any given sample point falling in A is

$$\mu P_1(A) + (1-\mu)P_2(A) = P_3(A)$$

Since N_1 samples are drawn from P_1 and N_2 samples from P_2, these samples collectively follow the distribution P_3 as each individual sample's probability of being in any measurable set A is consistent with $P_3(A)$. Therefore, drawing $N_1 + N_2$ samples in this manner is equivalent to drawing $N_1 + N_2$ samples from P_3.

B.2 Proof of Proposition 1

Before procedding to the proof, we first establish a useful lemma.

Lemma 1. *If Assumption 1 holds, then for any $h \in \mathcal{H}$, we have*

$$\mathcal{L}_{ntp}(P^m, h) = \alpha \mathcal{L}_{ntp}(P^n, h) + (1-\alpha)\mathcal{L}_{ntp}(P^c, h)$$

Proof. Let x_i, $i = 1, 2, \ldots, |\mathcal{X}|$ denote all prefixes, and w_j, $j = 1, 2, \ldots, V$ denote all tokens. For all $x \in \mathcal{X}$, by Equation (1), we have:

$$\begin{aligned} P^m_X(x) &= \sum_{j=1}^{V} P^m(x, w_j) = \sum_{j=1}^{V} \alpha P^n(x, w_j) + (1-\alpha)P^c(x, w_j) \\ &= \alpha \sum_{j=1}^{V} P^n(x, w_j) + (1-\alpha)\sum_{j=1}^{V} P^c(x, w_j) = \alpha P^n_X(x) + (1-\alpha)P^c_X(x) \qquad (7) \end{aligned}$$

This indicates that the marginal distribution possesses additivity. Consequently,

$$\begin{aligned} \mathcal{L}_{ntp}(P^m, h) &= \mathbb{E}_{x\sim P^m_X}\mathbb{E}_{w\sim P^m_{\cdot|x}} - \log(\boldsymbol{p}^h_{\cdot|x}(w)) = \sum_{i=1}^{|\mathcal{X}|} P^m_X(x_i) \cdot \mathbb{E}_{w\sim P^m_{\cdot|x_i}} - \log(\boldsymbol{p}^h_{\cdot|x_i}(w)) \\ &= \sum_{i=1}^{|\mathcal{X}|} [(1-\alpha)P^c_X(x_i) + \alpha P^n_X(x_i)] \cdot \mathbb{E}_{w\sim P^m_{\cdot|x_i}} - \log(\boldsymbol{p}^h_{\cdot|x_i}(w)) \qquad \text{(Equation (7))} \\ &= (1-\alpha)\sum_{i=1}^{|\mathcal{X}|} P^c_X(x_i)\mathbb{E}_{w\sim P^m_{\cdot|x_i}} - \log(\boldsymbol{p}^h_{\cdot|x_i}(w)) + \alpha \sum_{i=1}^{|\mathcal{X}|} P^n_X(x_i)\mathbb{E}_{w\sim P^m_{\cdot|x_i}} - \log(\boldsymbol{p}^h_{\cdot|x_i}(w)) \\ & \qquad (8) \end{aligned}$$

The conditional distributions do not generally exhibit a linear relationship:

$$P^m_{\cdot|x}(w|x) = \frac{P^m(x, w)}{P^m_X(x)} = \frac{(1-\alpha)P^c(x, w) + \alpha P^n(x, w)}{(1-\alpha)P^c_X(x) + \alpha P^n_X(x)} \neq P^c_{\cdot|x}(w|x) \neq P^n_{\cdot|x}(w|x)$$

However, if $\text{supp}(P^c) \cap \text{supp}(P^n) = \emptyset$, it immediately follows that:

$$P^m_{\cdot|x}(w|x) = \frac{(1-\alpha)P^c(x, w) + \alpha P^n(x, w)}{(1-\alpha)P^c_X(x) + \alpha P^n_X(x)} = \begin{cases} P^c_{\cdot|x}(w|x) & \text{if } (x, w) \in \text{supp}(P^c), \\ P^n_{\cdot|x}(w|x) & \text{if } (x, w) \in \text{supp}(P^n). \end{cases}$$

Consequently,

$$\sum_{i=1}^{|\mathcal{X}|} P^c_X(x_i)\mathbb{E}_{w\sim P^m_{\cdot|x_i}} - \log(\boldsymbol{p}^h_{\cdot|x_i}(w)) = \sum_{i=1}^{|\mathcal{X}|} P^c_X(x_i)\mathbb{E}_{w\sim P^c_{\cdot|x_i}} - \log(\boldsymbol{p}^h_{\cdot|x_i}(w)) = \mathcal{L}_{ntp}(P^c, h) \qquad (9)$$

Similarly,

$$\sum_{i=1}^{|\mathcal{X}|} P^n_X(x_i)\mathbb{E}_{w\sim P^m_{\cdot|x_i}} - \log(\boldsymbol{p}^h_{\cdot|x_i}(w)) = \mathcal{L}_{ntp}(P^n, h) \qquad (10)$$

By substituting Equation (9) and Equation (10) into Equation (8), the proof is completed. □

Now we can prove Proposition 1.

Proposition 1. *Under Assumption 1, let h^* be a model trained on P^c, with $\mathcal{L}_{ntp}(P^c, h^*) = -\log p_c$ and $\mathcal{L}_{ntp}(P^n, h^*) = -\log p_n$. When the model h is trained on a mixed distribution P^m which includes noise, it attempts to fit P^n, leading to an increase in the loss on the clean distribution P^c, such that $\mathcal{L}_{ntp}(P^c, h) = -\log(p_c - \epsilon)$ and $\mathcal{L}_{ntp}(P^n, h) = -\log(p_n + \epsilon/k)$ for some $\epsilon > 0$ (k can be shown to be $\Omega(e^{\mathcal{L}_{ntp}(P^n,h)})$). Let $\eta = \alpha p_c - (1-\alpha)kp_n$. We arrive at the following conclusions:*

(1) If $\alpha \leq \frac{kp_n}{p_c+kp_n}$, then for any $0 < \epsilon < p_c$, we have $\mathcal{L}_{ntp}(P^m, h) \geq \mathcal{L}_{ntp}(P^m, h^*)$. This means that when α is sufficiently small, the global minimum on P^m will not be affected by noise.

(2) If $\alpha > \frac{kp_n}{p_c+kp_n}$, then for $\epsilon < \eta$, it holds that $\mathcal{L}_{ntp}(P^m, h) < \mathcal{L}_{ntp}(P^m, h^*)$. This suggests that if α is large enough, the impact on the optimal hypothesis is at least as much as $\alpha p_c - (1-\alpha)kp_n$.

(3) When $\alpha < \frac{1}{3}$ and $k > \frac{\alpha(1-3\alpha)p_c}{(1-\alpha)(2-3\alpha)p_n}$, for $\epsilon \geq 3\eta$ we get $\mathcal{L}_{ntp}(P^m, h^*) < \mathcal{L}_{ntp}(P^m, h)$. Similarly, it can be shown that ϵ does not exceed 2η when $\alpha > \max(\frac{kp_n}{p_c+kp_n}, \frac{1}{2})$ and $k > \frac{(2\alpha-1)p_c}{2(1-\alpha)p_n}$. This indicates that when k is sufficiently large, the effect of noise is at most $\mathcal{O}(\alpha p_c - (1-\alpha)kp_n)$.

Proof. We first establish that k is $\Omega(e^{\mathcal{L}_{ntp}(P^n,h)})$, thereby ensuring that $\eta \ll \alpha p_c$. Note that

$$\epsilon = \frac{1}{e^{\mathcal{L}_{ntp}(P^c,h)}} - \frac{1}{e^{\mathcal{L}_{ntp}(P^c,h)}} = \frac{e^{\mathcal{L}_{ntp}(P^c,h)-\mathcal{L}_{ntp}(P^c,h)} - 1}{e^{\mathcal{L}_{ntp}(P^c,h)}} \tag{11}$$

$$\frac{\epsilon}{k} = \frac{1}{e^{\mathcal{L}_{ntp}(P^n,h)}} - \frac{1}{e^{\mathcal{L}_{ntp}(P^n,h)}} = \frac{e^{\mathcal{L}_{ntp}(P^n,h)-\mathcal{L}_{ntp}(P^n,h)} - 1}{e^{\mathcal{L}_{ntp}(P^n,h)}} \tag{12}$$

Therefore

$$\begin{aligned} k = \frac{\epsilon}{\frac{\epsilon}{k}} &= e^{\mathcal{L}_{ntp}(P^n,h)-\mathcal{L}_{ntp}(P^c,h)} \cdot \frac{e^{\mathcal{L}_{ntp}(P^c,h)-\mathcal{L}_{ntp}(P^c,h)} - 1}{e^{\mathcal{L}_{ntp}(P^n,h)-\mathcal{L}_{ntp}(P^n,h)} - 1} \\ &> e^{\mathcal{L}_{ntp}(P^n,h)-\mathcal{L}_{ntp}(P^c,h)} \cdot \frac{\mathcal{L}_{ntp}(P^c, h) - \mathcal{L}_{ntp}(P^c, h)}{e^{\mathcal{L}_{ntp}(P^n,h)-\mathcal{L}_{ntp}(P^n,h)}} \\ &= e^{\mathcal{L}_{ntp}(P^n,h)} \cdot \frac{\mathcal{L}_{ntp}(P^c, h) - \mathcal{L}_{ntp}(P^c, h)}{e^{\mathcal{L}_{ntp}(P^c,h)}} \end{aligned} \tag{13}$$

where $\frac{\mathcal{L}_{ntp}(P^c,h)-\mathcal{L}_{ntp}(P^c,h)}{e^{\mathcal{L}_{ntp}(P^c,h)}}$ only depends on P^c, h and h. It is worth noting that when P^n is random noise, $e^{\mathcal{L}_{ntp}(P^n,h)-\mathcal{L}_{ntp}(P^n,h)} - 1$ is close to 0, which leads to k exceeding the lower bound established in Equation (13). Then:

(1) If $\alpha \leq \frac{kp_n}{p_c+kp_n}$, we have:

$$\begin{aligned} \mathcal{L}_{ntp}(P^m, h^*) - \mathcal{L}_{ntp}(P^m, h) &= (1-\alpha)(\mathcal{L}_{ntp}(P^c, h^*) - \mathcal{L}_{ntp}(P^c, h)) + \alpha(\mathcal{L}_{ntp}(P^n, h^*) - \mathcal{L}_{ntp}(P^n, h)) \\ &= (1-\alpha)\log\frac{p_c - \epsilon}{p_c} + \alpha\log\frac{p_n + \frac{\epsilon}{k}}{p_n} && (14) \\ &\leq (1-\alpha)\cdot\frac{-\epsilon}{p_c} + \alpha\cdot\frac{\frac{\epsilon}{k}}{p_n} && (\log(1+t) \leq t) \\ &= \epsilon[\frac{(\alpha-1)}{p_c} + \frac{\alpha}{kp_n}] = \epsilon\frac{\alpha p_c - (1-\alpha)kp_n}{kp_cp_n} && (15) \end{aligned}$$

As $\alpha \leq \frac{kp_n}{p_c+kp_n} \iff \alpha p_c - (1-\alpha)kp_n \leq 0$, for $\epsilon > 0$ we have $\mathcal{L}_{ntp}(P^m, h^*) \leq \mathcal{L}_{ntp}(P^m, h)$.

(2)when $\alpha > \frac{kp_n}{p_c+kp_n}$ and $\epsilon < \alpha p_c - (1-\alpha)kp_n$, we have

$$\begin{aligned}
\mathcal{L}_{ntp}(P^m, h^*) - \mathcal{L}_{ntp}(P^m, h) &= (1-\alpha)\log\frac{p_c-\epsilon}{p_c} + \alpha\log\frac{p_n+\frac{\epsilon}{k}}{p_n} && \text{(Equation (14))} \\
&\geq (1-\alpha)\frac{-\epsilon}{p_c-\epsilon} + \alpha\frac{\frac{\epsilon}{k}}{p_n+\frac{\epsilon}{k}} && (\log t \geq 1-\tfrac{1}{t}) \\
&= \epsilon(\frac{\alpha-1}{p_c-\epsilon} + \frac{\alpha}{kp_n+\epsilon}) \\
&= \frac{\epsilon}{(p_c-\epsilon)(kp_n+\epsilon)}[\alpha(p_c-\epsilon) - (1-\alpha)(kp_n+\epsilon)] \\
&= \frac{\epsilon}{(p_c-\epsilon)(kp_n+\epsilon)}[\alpha p_c - (1-\alpha)kp_n - \epsilon] && (16)
\end{aligned}$$

As $\epsilon < \alpha p_c - (1-\alpha)kp_n < \alpha p_c < p_c$, by Equation (16) we have $\mathcal{L}_{ntp}(P^m, h) - \mathcal{L}_{ntp}(P^m, h) > 0$.

(3) Let

$$f(\epsilon) = (1-\alpha)\log\frac{p_c-\epsilon}{p_c} + \alpha\log\frac{p_n+\frac{\epsilon}{k}}{p_n} \overset{p'_n=kp_n}{=} (1-\alpha)\log\frac{p_c-\epsilon}{p_c} + \alpha\log\frac{p'_n+\epsilon}{p'_n} \tag{17}$$

Take the derivative of $f(\epsilon)$:

$$f'(\epsilon) = (1-\alpha)\frac{-\frac{1}{p_c}}{1-\frac{\epsilon}{p_c}} + \alpha\frac{\frac{1}{p'_n}}{1+\frac{\epsilon}{p'_n}} = (1-\alpha)\frac{1}{\epsilon-p_c} + \alpha\frac{1}{p'_n+\epsilon} = \frac{[\alpha p_c - (1-\alpha)p'_n] - \epsilon}{(p_c-\epsilon)(p'_n+\epsilon)} \tag{18}$$

Without loss of generality, assume $\eta > 0$, then $f(\epsilon)$ is monotonically increasing on $[0, \eta)$ and monotonically decreasing on (η, p_c). Therefore, to prove that $\mathcal{L}_{ntp}(P^m, h^*) < \mathcal{L}_{ntp}(P^m, h)$ for $\epsilon \geq 3\eta$, we only need to show $f(3\eta) < 0$ when $k > \frac{\alpha(1-3\alpha)p_c}{(1-\alpha)(2-3\alpha)p_n}$. Notice that

$$\begin{aligned}
f(3\eta) &= (1-\alpha)\log(1 - \frac{3\alpha p_c - 3(1-\alpha)p'_n}{p_c}) + \alpha\log(1 + \frac{3\alpha p_c - 3(1-\alpha)p'_n}{p'_n}) \\
&= (1-\alpha)\log(1 - 3\alpha + \frac{3(1-\alpha)}{\frac{p_c}{p'_n}}) + \alpha\log(3\alpha - 2 + 3\alpha\frac{p_c}{p'_n}) && (19)
\end{aligned}$$

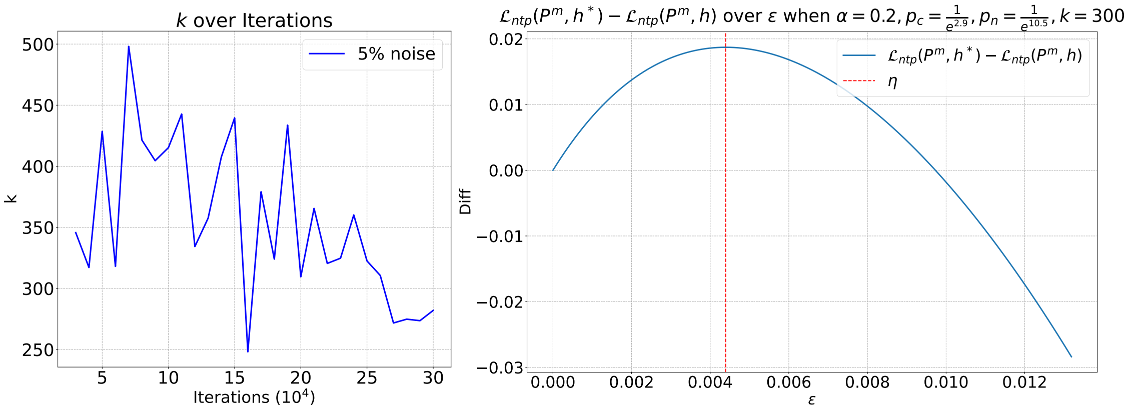
Let

$$g_3(t) = (1-\alpha)\log(1 - 3\alpha + \frac{3(1-\alpha)}{t}) + \alpha\log(3\alpha - 2 + 3\alpha t) \tag{20}$$

Take the derivative:

$$\begin{aligned}
g'_3(t) &= (1-\alpha)\frac{1}{1-3\alpha+\frac{3(1-\alpha)}{t}}\frac{3(\alpha-1)}{t^2} + \alpha\frac{3\alpha}{3\alpha-2+3\alpha t} && (21) \\
&= \frac{-3(1-\alpha)^2}{(1-3\alpha)t^2 + (1-\alpha)t} + \frac{3\alpha^2}{3\alpha-2+3\alpha t} && (22) \\
&= \frac{-3(1-\alpha)^2(3\alpha-2+3\alpha t) + 3\alpha^2[(1-3\alpha)t^2 + (1-\alpha)t]}{[(1-3\alpha)t^2 + (1-\alpha)t](3\alpha-2+3\alpha t)} && (23) \\
&= \frac{[\alpha t + (\alpha-1)][3\alpha(1-3\alpha)t + 3(1-\alpha)(3\alpha-2)]}{[(1-3\alpha)t^2 + (1-\alpha)t](3\alpha-2+3\alpha t)} && (24)
\end{aligned}$$

First, consider the denominator. Since $\alpha < \frac{1}{3}$, it is clear that $(1-3\alpha)t^2 + (1-\alpha)t > 0$. Given that $t = \frac{p_c}{p'_n} > \frac{1-\alpha}{\alpha}$ (because $\eta > 0$), it follows that $3\alpha - 2 + 3\alpha t > 1 > 0$. Therefore, the denominator is always positive. Next, we consider the numerator. Since $\eta > 0$, it follows that $\alpha t + (\alpha - 1) > 0$. Therefore, when $t = \frac{p_c}{p'_n} = \frac{p_c}{kp_n} < \frac{(1-\alpha)(2-3\alpha)}{\alpha(1-3\alpha)}$, we have $g'_3(t) < 0$. This means that $g_3(t)$ is monotonically decreasing on $(\frac{1-\alpha}{\alpha}, \frac{(1-\alpha)(2-3\alpha)}{\alpha(1-3\alpha)}]$. Consequently, $f(3\eta) = g_3(t) \leq g_3\left(\frac{1-\alpha}{\alpha}\right) = 0$.



(a)

(b)

Figure 7: Visualization of k and $\mathcal{L}_{ntp}(P^m, h^*) - \mathcal{L}_{ntp}(P^m, h)$. (a) The trend of k as it changes with training, plotted using the model trained on P^c as h^*. (b) Visualization of $\mathcal{L}_{ntp}(P^m, h)$ when the parameter settings are consistent with the experiment.

Following the same line of reasoning, when $\alpha > \frac{1}{2}$, we have

$$f(2\eta) = (1-\alpha)\log(1 - \frac{2\alpha p_c - 2(1-\alpha)p_n'}{p_c}) + \alpha\log(1 + \frac{2\alpha p_c - 2(1-\alpha)p_n'}{p_n'})$$

$$= (1-\alpha)\log(1 - 2\alpha + \frac{2(1-\alpha)}{\frac{p_c}{p_n'}}) + \alpha\log(2\alpha - 1 + 2\alpha\frac{p_c}{p_n'}) \tag{25}$$

Let

$$g_2(t) = (1-\alpha)\log(1 - 2\alpha + \frac{2(1-\alpha)}{t}) + \alpha\log(2\alpha - 1 + 2\alpha t) \tag{26}$$

Take the derivative:

$$g_2'(t) = (1-\alpha)\frac{1}{1 - 2\alpha + \frac{2(1-\alpha)}{t}}\frac{2(\alpha-1)}{t^2} + \alpha\frac{2\alpha}{2\alpha - 1 + 2\alpha t} \tag{27}$$

$$= \frac{-2(1-\alpha)^2}{(1-2\alpha)t^2 + 2(1-\alpha)t} + \frac{2\alpha^2}{2\alpha - 1 + 2\alpha t} \tag{28}$$

$$= \frac{2(1-2\alpha)(\alpha t + 1 - \alpha)^2}{[(1-2\alpha)t^2 + 2(1-\alpha)t](2\alpha - 1 + 2\alpha t)} \tag{29}$$

Therefore, when $\frac{1-\alpha}{\alpha} < t < \frac{2(1-\alpha)}{2\alpha-1}$, we have $g_2'(t) < 0$, which implies that $f(2\eta) < 0$.

□

B.3 Justification of Proposition 1

We plot the trend of k in Figure 7(a). We compare checkpoints trained for the same iterations on both P^c and P^m, where p_c is calculated based on the loss of the model trained on P^c, and p_n is determined by the loss of a model trained for 10,000 iterations on P^m when evaluated on P^n. It can be observed that the value of k corresponding to random noise is significantly greater than one, which supports the rationality of the assumption made in Proposition 1.

On the other hand, to extend the proposed theory beyond uniformly distributed random noise (for instance, in multilingual models or Gaussian noise), it is necessary to ensure that k does not become too small in these scenarios. This means that $\mathcal{L}_{ntp}(P^n, h^*) = -\log p_n$ should not be close to $\log V$. One trivial way to increase p_n is to decrease V, the size of vocabulary. Apart from this, we provide two lines of reasoning to justify why p_n can be made large:

(1) Numerous studies on compressing large language models, such as pruning (Wang et al., 2020; Kurtic et al., 2022; Zhang et al., 2024), quantization (Zhao et al., 2024; Jin et al., 2024; Liu et al., 2024),

and distillation (Dasgupta et al., 2023; Hinton, 2015), have demonstrated that there exists a significant amount of redundancy within the parameters of large models. Therefore, we could first train a model on P^c and then compress it, fine-tuning the surplus parameters on P^n. This approach would allow us to improve p_n without altering p_c.

(2) A small proportion of data corresponding to P^n can be introduced into P^c, making sure that α is extremely small. According to domain adaptation theory (Ben-David et al., 2010), this would only slightly increase $\mathcal{L}_{ntp}(P^c)$. However, existing results (Shliazhko et al., 2024; Pires et al., 2019; Chi et al., 2020) indicate that pre-trained models like BERT or GPT on English text can exhibit strong multilingual capabilities with just a very limited amount of data. Consequently, compared to a model trained solely on P^c, the resulting model has a minor difference in p_c but a relatively higher p_n.

Both thought experiments above demonstrate that there exist a lot of models within the parameter space $\mathcal{H}$ can perform well on P^c while yielding non-trivial outcomes on P^n. Thus, we can ensure that models trained on mixed data distributions will have a sufficiently large k.

Additionally, in Figure 7(b), we illustrate how $\mathcal{L}_{ntp}(P^m)$ varies with changes in ϵ, under settings identical to those used during pre-training. The results depicted in the figure are consistent with our theoretical derivations.

B.4 Omitted Details in Section 4.2

Definition 1 (β-smooth (Zheng et al., 2023b))**.** *A loss function $\ell(g_\theta(t), y)$ is β-smooth, if for any $(t, y) \in \mathcal{T} \times \mathcal{Y}$ and any $\theta, \theta' \in \Theta$,*

$$||\nabla_\theta \ell(g_\theta(t), y) - \nabla_{\theta'} \ell(g_{\theta'}(t), y)||_2 \leq \beta ||\theta - \theta'||_2 \tag{30}$$

Definition 2 (ρ-input flatness)**.** *The ρ-input flatness $R_\rho(\theta)$ of loss function $\ell(g_\theta(t), y)$ is defined as:*

$$R_\rho(\theta) = \mathbb{E}_{(t,y)\sim\mathcal{D}} \sup_{\delta' \in B(0,\rho)} \ell(g_\theta(t + \delta'), y) - \ell(g_\theta(t), y) \tag{31}$$

where $B(0, \rho) = \{\delta' : ||\delta'||_2 < \rho\}$ is a open ball.

Lemma 2. *If the loss function $\ell(g_\theta(t), y)$ is β-smooth, then*

$$||\nabla_\theta \ell(g_\theta(t), y)||_2^2 \leq 4\beta \ell(g_\theta(t), y) \tag{32}$$

Proof. See Lemma 3.1 in Srebro et al. (2010). □

Proposition 2. *Suppose $\ell(g_\theta(t), y)$ is β-smooth with ρ-input flatness $R_\rho(\theta)$, for any $\theta \in \Theta$:*

$$\mathcal{L}_{gm}(\theta) \leq 2\beta + 2\mathcal{L}_{ce}(\mathcal{D}, g_\theta) + R_\rho(\theta) \tag{33}$$

Proof.

$$\begin{aligned}
\mathcal{L}_{gm}(\theta) &= ||\mathbb{E}_{(t,y)\sim\mathcal{D}} \nabla_\theta \ell(g_\theta(t), y) - \mathbb{E}_{(\hat{t},y)\sim\hat{\mathcal{D}}} \nabla_\theta \ell(g_\theta(\hat{t}), y)||_2 && (34) \\
&\leq ||\mathbb{E}_{(t,y)\sim\mathcal{D}} \nabla_\theta \ell(g_\theta(t), y)||_2 + ||\mathbb{E}_{(\hat{t},y)\sim\hat{\mathcal{D}}} \nabla_\theta \ell(g_\theta(\hat{t}), y)||_2 && \text{(Triangle Inequality)} \\
&\leq \mathbb{E}_{(t,y)\sim\mathcal{D}} ||\nabla_\theta \ell(g_\theta(t), y)||_2 + \mathbb{E}_{(\hat{t},y)\sim\hat{\mathcal{D}}} ||\nabla_\theta \ell(g_\theta(\hat{t}), y)||_2 && \text{(Jensen's Inequality)} \\
&\leq \mathbb{E}_{(t,y)\sim\mathcal{D}} 2\sqrt{\beta \ell(g_\theta(t), y)} + \mathbb{E}_{(\hat{t},y)\sim\hat{\mathcal{D}}} 2\sqrt{\beta \ell(g_\theta(\hat{t}), y)} && \text{(Lemma 2)} \\
&\leq \mathbb{E}_{(t,y)\sim\mathcal{D}} (\beta + \ell(g_\theta(t), y)) + \mathbb{E}_{(\hat{t},y)\sim\hat{\mathcal{D}}} (\beta + \ell(g_\theta(\hat{t}), y)) && \text{(AM-GM Inequality)} \\
&= 2\beta + 2\mathbb{E}_{(t,y)\sim\mathcal{D}} \ell(g_\theta(t), y) + (\mathbb{E}_{(\hat{t},y)\sim\hat{\mathcal{D}}} \ell(g_\theta(\hat{t}), y) - \mathbb{E}_{(t,y)\sim\mathcal{D}} \ell(g_\theta(t), y)) && (35) \\
&\leq 2\beta + 2\mathcal{L}_{ce}(\mathcal{D}, g_\theta) + R_\rho(\theta) && (36)
\end{aligned}$$

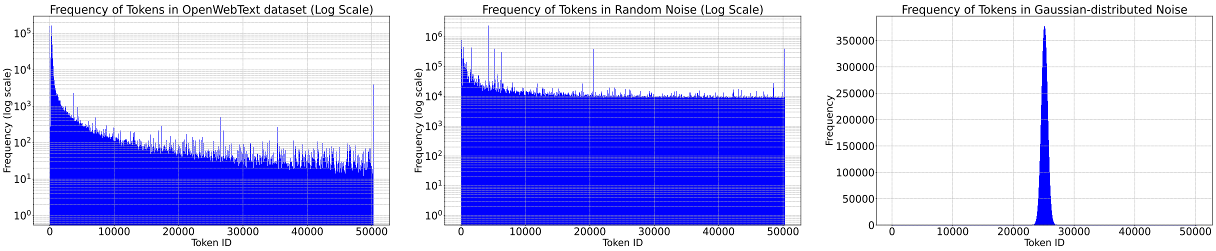
where the last inequality holds because $\hat{t} - t \in B(0, \rho)$. □

C Detailed Related Works

Data selection for language model training. LLMs (Yang et al., 2023c; Zhuang et al., 2024c,d; Xie et al., 2024; Pan et al., 2024; Weng et al., 2025; Fan et al., 2024c) have fundamentally change the landsape of current AI research (Pan et al., 2023; Fan et al., 2024a; Zhuang et al., 2024a,e; Yang et al., 2023a, 2024b). Large text corpora form the backbone of language models, with data quality being fundamental to their success and safety (Yang et al., 2023e,f). Elazar et al. (2024) conducted a systematic analysis of open-source text datasets such as The Pile (Gao et al., 2020) (used to train Pythia), C4 (Raffel et al., 2020) (used to train T5) and RedPajama (used to train LLaMA), revealing that they contain a significant amount of duplicate, toxic, synthetic, and low-quality content. Therefore, it is of great importance to thoroughly understand the impact of low-quality data within these pre-training datasets on the model's performance, reliability, and safety. Allen-Zhu and Li (2024a,b) systematically investigated the effect of low-quality data and found that such data can significantly reduce the model's knowledge capacity, sometimes by up to 20 times. Another research direction primarily focuses on the synthetic data of large language models, specifically examining the impacts of using data generated by LLMs for recursive training. The study by Shumailov et al. (2024) was the first to explore this issue and introduced the concept of "model collapse", indicating that recursive training can lead to the loss of information in tail tokens, ultimately resulting in the model producing nonsensical content. Seddik et al. (2024) mainly provided a theoretical explanation for why model collapse occurs, supporting their arguments with experimental evidence. Consequently, the importance of data selection cannot be overstated. Given that data selection is an NP-hard problem in terms of combinatorial optimization (Xiao et al., 2022), numerous heuristic algorithms have been proposed to expedite the process. Longpre et al. (2024) provided a comprehensive study on pre-training data selection and optimal ratios, offering practical recommendations. Yang et al. (2023d) proposed dataset pruning, an approach that assesses the impact of omitting training samples on model generalization and creates a minimal training subset with a controlled generalization gap. Chai et al. (2024) evaluated the impact of individual training samples on the dynamics of GPT model training. Li et al. (2024b) introduced the Instruction-Following Difficulty metric to assess the quality of instruction-tuning data. Xie et al. (2023b) employed importance resampling for data selection. Xie et al. (2023a); Lee et al. (2023) advocated for optimizing data composition and diversity. Despite these notable studies on data selection, they generally acknowledge that dataset noise degenerates model performance but lack a detailed understanding of how and to what extent, particularly in the case of random noise which is inevitable in large-scale datasets. Although Cherepanova and Zou (2024) investigated the influence of gibberish input, the random noise within the pre-training dataset is still underexplored. This paper aims to bridge the gap.

Learning from Noisy Distributions. The majority of machine learning algorithms assume that training and test samples are independently and identically distributed (i.i.d.), a condition that is often not met in real-world scenarios. For instance, LLMs are pre-trained on datasets with all kinds of noise while their performance is evaluated by the user whose distribution is usually clean and meets real-world scenarios, which violates the i.i.d. assumption. Domain adaptation (Li et al., 2024a; Meng et al., 2022; Ma et al., 2023) addresses this issue when the distribution of the training data differs from that of the test data. Although domain adaptation methods attempt to reduce the statistical distribution discrepancy (Du and Li, 2023; Wu et al., 2024) or employ adversarial training (Li et al., 2021; Ru et al., 2024) to minimize the gap between source and target domains, they typically require access to unlabeled test data under a semi-supervised learning setup, which is impractical for LLM training. Another reason domain adaptation cannot be directly applied here is that domain adaptation theory (Ben-David et al., 2010) focuses on the performance of a model trained on one distribution when it is applied to another different but related distribution. This kind of bounds can be easily derived by Lemma 1. However, what we aim to investigate here is the extent of performance loss when comparing a model trained on one distribution (noisy dataset) to a model trained on another distribution (clean dataset).

Apart from domain adaptation, there has been extensive research directly investigating noisy training sets. Noisy label learning Song et al. (2022); Lukasik et al. (2020) have explored the impact of incorrect labels on model performance. Regarding input feature noise, Smilkov et al. (2017) added perturbations



(a)

(b)

(c)

Figure 8: The prior distribution of tokens in the data from (a) OpenWebText, (b) random noise, and (c) Gaussian noise.

to individual image inputs to enhance model interpretability, and Zada et al. (2022) added white noise image into the training dataset to tackle the class imbalance problem. However, most of these efforts have concentrated on image classification and do not consider the pre-training paradigm.

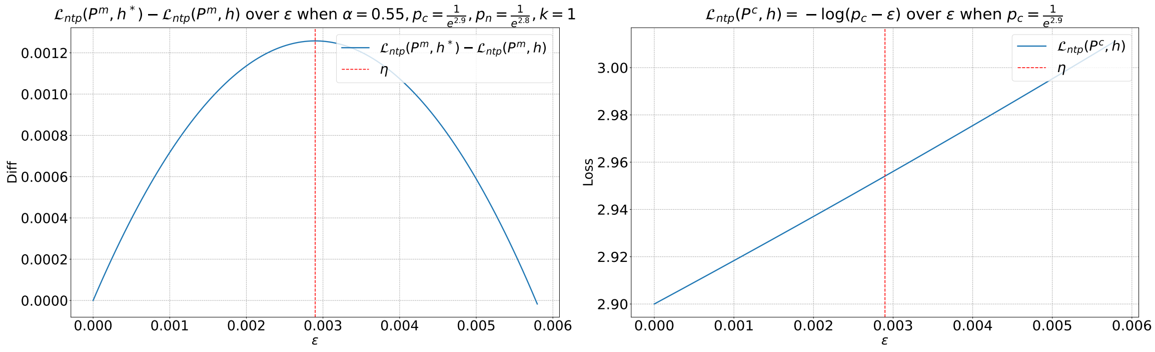
Fine-tuning Pre-trained Models. The approach of initially pre-training model weights on large-scale datasets and subsequently fine-tuning them with downstream data has become the de facto standard in the fields of computer vision (Ye and Gao, 2024; Zhuang et al., 2025b; Fan et al., 2024b; Zhuang et al., 2025a), natural language processing (Wei et al., 2021; Xie et al., 2021a; Zhuang et al., 2024f; Tang et al., 2024) and audio processing (Yang et al., 2023b; Zhuang et al., 2024b, 2025c). For instance, Hua et al. (2023, 2021) proposed enhancing the performance of models by increasing their resistance to minor perturbations in intermediate layers. Meanwhile, Jiang et al. (2020) improved model robustness by adding regularization terms. Besides full-parameter fine-tuning, numerous parameter-efficient fine-tuning algorithms have been extensively studied. Zhang et al. (2021) introduced adapters into the original model architecture, optimizing only these parameters during fine-tuning. Zhou et al. (2022b,a) efficiently fine-tuned CLIP models (Radford et al., 2021) using learnable soft prompts. Hu et al. (2022) optimized models through learning low-rank residual weights. These methods achieved performance close to that of full-parameter fine-tuning while maintaining the generalization ability of the original models. However, they all require access to the model's weights and loading them into GPU memory, which can be challenging for today's large models, especially when state-of-the-art models' parameters are not publicly available. Therefore, in this paper, we follow the NML setup and explore efficient ways to fine-tune the downstream task head under a black-box scenario.

Implicit Regularization and Sharpness-aware Minimization. Achieving good generalization in neural networks optimized using gradient descent algorithms has long been a research focus in deep learning theory. Barrett and Dherin (2021) explored the properties of stochastic gradient descent (SGD), finding that SGD implicitly constrains the gradient norm. Based on this observation, Sharpness-aware minimization (SAM) (Zhang et al., 2023; Foret et al., 2021; Wen et al., 2023; Xie et al., 2023c) improves generalization by incorporating the gradient norm as a regularization term. Our method can be seen as drawing inspiration from SAM but differs in that our optimization objective is the model's resilience to input noise rather than seeking flat minima in the parameter space.

D Experiments in Section 3

D.1 Pre-training Dataset

OpenWebText Dataset. The OpenWebText dataset (Gokaslan et al., 2019) is a large-scale corpus of English text data, developed to serve as an open-access alternative to proprietary dataset WebText that is utilized by OpenAI for training their GPT-2 models. This dataset originates from the analysis of outbound links clicked on Reddit, undergoing multiple stages of filtering to exclude non-English content, duplicate entries, copyrighted materials, and texts lacking in quality. These links generally direct to web pages available to the public, often shared or debated on Reddit, thereby covering a broad spectrum of subjects that mirror online popular interests and discussions. The dataset includes roughly 18 million documents, amounting to about 20GB of compressed plain text data in uint16 format. Since measures have been



(a)

(b)

Figure 9: Visualization of (a) $\mathcal{L}_{ntp}(P^m, h^*) - \mathcal{L}_{ntp}(P^m, h)$ and (b) $\mathcal{L}_{ntp}(P^c, h)$ with $\alpha = 0.55, k = 1, \mathcal{L}_{ntp}(P^c, h^*) = 2.9, \mathcal{L}_{ntp}(P^n, h^*) = 2.8$.

implemented to ensure the dataset's reliability by filtering out unsuitable content, we consider it a clean and noise-free dataset. Figure 8(a) illustrates the distribution of internal tokens.

Random Noise. To simulate the distribution of random gibberish that crawlers might retrieve from the Internet due to various reasons, we manually searched and collected a few websites containing such gibberish and also opened normally functioning websites using different decoding methods to observe the distribution of tokens. We found that, while the distribution of tokens appeared disorganized, their prior probabilities were not evenly distributed. Instead, several tokens had notably high probabilities, which is similar to that observed in the clean data as shown in Figure 8(a). Thus, on the basis of uniformly distributed random noise, we increased the frequency of certain tokens and then randomized them again. The resulting distribution is illustrated in Figure 8(b). It can be seen that while maintaining an overall uniform distribution, the frequency of tokens with IDs ranging from 0 to 1000 is higher which closely mirrors real-world scenarios.

Gaussian Noise. Given the diversity and unpredictability of real-world data distributions, we also artificially generated random noise with a prior probability that follows a Gaussian distribution, as shown in Figure 8(c). The rationale behind choosing the Gaussian distribution is that the noise in many real-world systems can be approximated by it. Additionally, we set the standard deviation $\sigma = 500$ to simulate scenarios where random noise exhibits sharp peaks.

D.2 Training Details of GPT-2

Our work is based on the source code of nanoGPT[4]. Specifically, we utilized the GPT-2 tokenizer with vocabulary size $V = 50256$ to process the OpenWebText dataset, and then appended randomly generated noise to the end of the training set before commencing training. The model's context length is set to $L = 1024$, with an embedding dimension $d = 768$. The GPT-2 model consists of 12 self-attention layers, totaling approximately 124 million parameters. For optimization, we employed AdamW (Loshchilov, 2017; Xie et al., 2022) with a learning rate of 6e-4, weight decay of 0.1, and β values of 0.9 and 0.95 for β_1 and β_2, respectively. A cosine annealing scheduler was used to gradually adjust the learning rate down to 6e-5. We configured the batch size to 16, with a gradient accumulation step of 40, allowing each iteration to process 655,360 tokens (16 * 40 * 1024). Training proceeded for a total of 300,000 iterations.

D.3 Synthetic Results about Multilingual Models

To illustrate our theory's explanatory power concerning multilingual models, we have plotted the scenario where h^* is influenced by P^n under the conditions $\alpha = 0.55$, $k = 1$, $\mathcal{L}_{ntp}(P^c, h^*) = 2.9$, and $\mathcal{L}_{ntp}(P^n, h^*) = 2.8$, as shown in Figure 9. This setup simulates a model trained on a roughly 1:1 multilingual corpus, where the capacity of one language is affected by the data from another language. As can be observed from the figure, the impact on p_c does not exceed $2\eta = 2(\alpha p_c - (1 - \alpha)p_n)$, which

[4] https://github.com/karpathy/nanoGPT

translates to an increase of no more than 0.1 in $\mathcal{L}_{ntp}(P^n, h)$. This finding strongly supports the success of multilingual models from a theoretical perspective.

D.4 Hardware

We conducted the pre-training process on a server equipped with 8 NVIDIA GeForce RTX 4090 GPUs. It takes approximately 70 hours to train one model using eight 4090 GPUs, so pre-training five GPT-2 models in total requires 2,800 GPU hours.

D.5 Additional Results on GPT-2 774M and BERT-base

To further validate the impact of random noise, we scaled up the parameters of GPT-2 from 124M to 774M and conducted pre-training on a dataset containing 5% random noise. The results are presented in Table 8.

Noise Percentage	10,000 iters	50,000 iters	100,000 iters
0%	3.04	2.89	2.82
5%	3.06	2.92	2.83

Table 8: NTP loss during GPT-2 774M training

These findings sufficiently demonstrate that our conclusions can be extended to models with larger scales, as our theoretical framework does not hinge upon model performance metrics.

Additionally, experiments in Section 4 involve BERT and vision models to assess the generalizability of the proposed local gradient matching function. This investigation is somewhat tangential to our primary focus—the impact of random noise on training language models using autoregressive approaches. Nonetheless, we pre-trained the BERT-base model under 5% noise conditions. The outcomes are illustrated below. These results align well with our theoretical predictions.

Noise Percentage	1M iters	5m iters	10m iters
0%	3.21	2.94	2.76
5%	3.23	2.97	2.79

Table 9: NTP loss during BERT-base training.

For vision models, pre-training outcomes can be found in Chen et al. (2024). Specifically, the authors simulated label noise by randomly replacing ImageNet sample labels and tested on 14 downstream tasks. The slight decrease in accuracy further corroborates our hypothesis across different types of models and datasets, underscoring the robustness of our approach against various levels of noise.

E Experiments in Section 4

E.1 Detailed Setup for Downstream Natural Language Understanding Experiments

E.1.1 Datasets

We utilize 8 commonly-used text classification benchmark: SST-2, SST-fine, 20newsgroup, CR, BBC, Balanced COPA, MRPC, WiC. The detailed information can be found in Table 10.

E.1.2 Prompts

Since classification tasks can be processed as seq2seq tasks by adding prompts (Sutskever, 2014; Saunshi et al., 2021), we design a unique prompt for each dataset and task. This approach transforms the inputs into a format that large language models can process. The specific designs are shown in Table 11.

Dataset	Classes	Train Size	Test Size
SST-2 (Socher et al., 2013)	2	6.92k	1.82k
SST-fine (Chen and Manning, 2014)	5	8.54k	2.21k
20newsgroup (Zhang et al., 2019)	20	11.3k	7.53k
CR (Hu and Liu, 2004)	2	3.39k	376
BBC (Samuels and Mcgonical, 2020)	5	1.23k	1k
Balanced COPA (Kavumba et al., 2020)	2	1k	500
MRPC (Dolan and Brockett, 2005)	2	3.67k	1.73k
WiC (Pilehvar and Camacho-Collados, 2019)	2	5.43k	1.4k

Table 10: Details of the 8 natural language understanding dataset.

Dataset	Prompts
SST-2	{text} this movie is
SST-fine	{text} this movie is
20newsgroup	{text} This article is about
CR	{text} the sentiment is
BBC	Please classify the topic of the following news: {text} Answer:
Balanced COPA	Given the premise: {premise} Find the most plausible alternative for the {question}. Option 1: {choice1} Option 2: {choice2} Which option is more plausible?
MRPC	Sentence 1: {text1} Sentence 2: {text2} Is this a paraphrase?
WiC	Task: Determine if the word {phrase1} has the same meaning in the two sentences below. Sentence 1: {sentence1} Sentence 2: {sentence2} Your answer:

Table 11: Details of the prompts applied to each dataset.

E.1.3 Hyperparameters

For all experiments in Section 4, we utilize a two-layer MLP with hidden dimension equals to feature dimension and ReLU activation function.

For all experiments shown in Table 1, we set γ in Equation (3) to be 0.01 and λ in Equation (6) to be 0.15. Following the setup as described by Saunshi et al. (2021), for each dataset, we conduct a grid search on the validation set to identify the optimal learning rate and batch size. We train for a total of ten epochs with the learning rate ranging within {3e-4, 6e-4} and batch size options including {8, 12, 16, 32}. For samples without a designated validation set, we randomly select 10% of the training set samples to form a validation set for the purpose of selecting the best parameters.

For the experiments listed in Table 2, we set the batch size to 8 and the learning rate to 6e-4 for all linear probe tasks. For all MLP probe tasks, the learning rate is set to 1e-4. Regarding γ and λ, we conduct a grid search on the validation set to find the optimal values.

E.2 Detailed Setup for Downstream Vision Experiments

E.2.1 Datasets

We select 14 image classification datasets, which serve as a common benchmark for evaluating model performance in the vision community (Zhou et al., 2022b; Chen et al., 2024). Specific information about these 14 datasets is provided in Table 12.

E.2.2 Models

We use five pre-trained general-purpose visual backbone models that cover a variety of architectures, datasets, and training methods. Detailed information is provided in Table 13.

Dataset	Classes	Train Size	Test Size
StanfordCars (Krause et al., 2013)	196	8144	8041
Caltech101 (Fei-Fei et al., 2004a)	102	3060	6084
CIFAR-10 (Krizhevsky et al., 2009)	10	50000	10000
CIFAR-100 (Krizhevsky et al., 2009)	100	50000	10000
DTD (Cimpoi et al., 2014)	47	1880	1880
EuroSAT (Helber et al., 2019)	10	21600	5400
FGVCAircraft (Maji et al., 2013)	102	6667	3333
Flowers102 (Nilsback and Zisserman, 2008)	102	2040	6149
Food101 (Fei-Fei et al., 2004b)	101	75750	25250
OxfordPet (Parkhi et al., 2012)	37	3680	3669
PatchCamelyon (Veeling et al., 2018)	2	262144	32768
RESISC45 (Cheng et al., 2017)	45	25200	6300
Rendered SST2 (Socher et al., 2013)	2	6920	1821
SVHN (Netzer et al., 2011)	10	73257	26032

Table 12: Details of the 14 vision dataset.

Model	Pre-training Dataset	Size
EfficientNet-B3 (Tan and Le, 2019)	ImageNet-1K (Deng et al., 2009) and JFT-300M (Sun et al., 2017)	12.3M
ResNetv2-152x2 (He et al., 2016)	ImageNet-21K (Ridnik et al., 2021)	321.7M
Swin-L (Liu et al., 2021)	ImageNet-21K	196.7M
ConvNext-L (Woo et al., 2023)	Laion-2B (Schuhmann et al., 2022) and ImageNet-1K	200.1M
ViT-L (Dosovitskiy, 2020)	Laion-2B	428M

Table 13: Details of the 5 vision models.

E.2.3 Hyperparameters

In our study, similar to the approach detailed in Chen et al. (2024), we primarily contrast our proposed method with MLP and LP tuning. For the optimization process, we employ AdamW for fine-tuning the modules over 30 epochs, utilizing a cosine learning rate scheduler. Specifically, for LP, we configure the learning rate at 0.1 without applying any weight decay. In contrast, both the MLP tuning and our method use a more conservative learning rate of 0.001 alongside a weight decay of 1e-4.

E.2.4 Detailed Experimental Results

In Table 3, due to space limitations, we only present the average results, while detailed results are shown in Table 14.

Models	StanfordCars		Caltech101		CIFAR-10	
	Linear	MLP	Linear	MLP	Linear	MLP
ViT-L	93.38 ± 0.76	94.41 ± 1.05	92.07 ± 1.19	95.20 ± 1.12	97.99 ± 0.95	98.35 ± 0.95
ViT-L + $\mathcal{L}_{gm}$	**93.71 ± 1.37**	**94.56 ± 1.50**	**95.01 ± 0.97**	**95.29 ± 1.27**	**98.07 ± 0.74**	**98.48 ± 0.60**
ConvNext-L	86.01 ± 1.48	88.68 ± 0.89	91.02 ± 0.79	94.47 ± 0.53	97.49 ± 1.36	98.09 ± 0.85
ConvNext-L+$\mathcal{L}_{gm}$	**86.78 ± 1.32**	**89.06 ± 1.19**	**94.11 ± 1.19**	**94.93 ± 0.88**	**97.59 ± 0.52**	**98.15 ± 0.71**
EfficientNet-B3	56.20 ± 0.54	**58.57 ± 1.11**	89.43 ± 0.78	91.22 ± 1.23	94.04 ± 1.19	95.73 ± 1.07
EfficientNet-B3+$\mathcal{L}_{gm}$	**57.02 ± 1.28**	58.15 ± 1.12	**90.25 ± 1.43**	**91.55 ± 0.90**	**94.11 ± 0.88**	**95.96 ± 0.86**
ResNetv2-152x2	56.95 ± 1.21	**59.18 ± 1.34**	91.40 ± 1.47	92.48 ± 1.30	96.28 ± 1.16	96.91 ± 0.89
ResNetv2-152x2+$\mathcal{L}_{gm}$	**58.78 ± 1.16**	58.67 ± 1.27	**93.83 ± 0.91**	**93.95 ± 0.94**	**96.31 ± 0.85**	**97.03 ± 0.53**
Swin-L	68.17 ± 0.98	**74.11 ± 0.60**	92.58 ± 0.95	94.09 ± 1.04	98.26 ± 0.89	98.61 ± 0.78
Swin-L+$\mathcal{L}_{gm}$	**69.31 ± 1.07**	73.71 ± 0.94	**93.65 ± 1.42**	**94.62 ± 0.64**	**98.41 ± 0.91**	**98.72 ± 1.28**

CIFAR-100		EuroSAT		FGVCAircraft		OxfordPet	
Linear	MLP	Linear	MLP	Linear	MLP	Linear	MLP
88.07 ± 0.58	89.49 ± 0.52	97.53 ± 1.13	97.75 ± 0.61	65.76 ± 0.73	68.43 ± 0.78	91.65 ± 1.18	93.97 ± 1.50
88.06 ± 0.93	**89.58 ± 0.93**	**97.83 ± 0.73**	**98.03 ± 0.60**	**66.63 ± 1.08**	**68.67 ± 1.07**	**93.18 ± 0.81**	**94.17 ± 1.10**
86.76 ± 0.91	87.79 ± 1.27	95.57 ± 1.22	96.31 ± 1.11	57.18 ± 1.12	62.25 ± 0.66	94.98 ± 0.54	95.80 ± 0.75
86.46 ± 1.04	**87.88 ± 1.24**	**96.05 ± 1.46**	**96.74 ± 1.27**	**58.35 ± 0.68**	**63.61 ± 0.83**	**95.39 ± 1.21**	**95.99 ± 0.92**
77.34 ± 0.86	80.28 ± 1.02	94.81 ± 1.35	95.90 ± 0.88	44.73 ± 1.31	46.23 ± 0.92	93.84 ± 1.14	94.79 ± 1.14
77.16 ± 1.11	**80.47 ± 1.43**	**95.20 ± 0.52**	**96.07 ± 1.18**	**45.33 ± 0.61**	**47.07 ± 0.57**	**94.63 ± 1.03**	**94.98 ± 1.14**
84.30 ± 1.18	**84.68 ± 1.33**	97.12 ± 1.46	97.46 ± 1.28	42.03 ± 0.72	48.39 ± 0.85	91.93 ± 0.68	92.99 ± 1.40
84.28 ± 1.22	84.29 ± 1.38	**97.35 ± 1.12**	**97.59 ± 0.72**	**45.69 ± 0.80**	**48.84 ± 0.61**	**92.61 ± 0.87**	**93.45 ± 1.32**
89.68 ± 1.33	90.74 ± 0.98	**97.11 ± 0.72**	97.59 ± 0.63	54.96 ± 1.24	**61.17 ± 1.35**	92.17 ± 0.64	94.38 ± 1.21
89.79 ± 0.52	**91.18 ± 1.21**	97.09 ± 1.11	**97.71 ± 1.08**	**56.10 ± 0.67**	60.99 ± 0.73	**93.86 ± 0.85**	**94.57 ± 1.11**

Food101		Flowers102		DTD		SVHN	
Linear	MLP	Linear	MLP	Linear	MLP	Linear	MLP
90.51 ± 1.31	91.04 ± 1.35	94.04 ± 1.13	97.83 ± 0.74	80.53 ± 1.06	83.29 ± 0.86	78.82 ± 1.25	84.74 ± 1.08
90.62 ± 1.37	**91.23 ± 0.52**	**96.67 ± 1.41**	**98.06 ± 1.28**	**82.76 ± 0.71**	**83.77 ± 0.98**	**79.80 ± 0.88**	**84.59 ± 1.38**
89.09 ± 1.06	**90.21 ± 0.87**	94.71 ± 1.31	98.78 ± 1.17	76.01 ± 1.03	78.67 ± 1.15	66.16 ± 0.87	72.76 ± 0.78
88.62 ± 0.72	90.10 ± 1.39	**97.12 ± 0.95**	**98.99 ± 0.99**	**77.92 ± 0.72**	**80.05 ± 1.44**	**68.43 ± 1.37**	**73.18 ± 0.67**
76.95 ± 0.82	**81.78 ± 0.89**	84.19 ± 0.61	88.97 ± 1.32	69.09 ± 1.27	73.08 ± 1.30	54.26 ± 0.87	61.38 ± 0.82
76.35 ± 0.74	81.33 ± 1.45	**86.19 ± 1.08**	**89.42 ± 0.62**	**71.27 ± 1.24**	**73.82 ± 1.40**	**56.74 ± 0.73**	**63.56 ± 0.82**
84.83 ± 1.36	84.15 ± 1.27	96.76 ± 0.88	98.27 ± 0.53	72.23 ± 0.94	76.11 ± 1.25	60.75 ± 0.89	64.87 ± 1.45
84.41 ± 0.88	**84.64 ± 1.04**	**98.08 ± 1.23**	**98.84 ± 0.82**	**74.73 ± 1.38**	**77.12 ± 1.39**	**62.04 ± 1.01**	**65.06 ± 0.62**
90.23 ± 0.64	92.23 ± 0.55	97.28 ± 0.92	99.51 ± 1.17	75.85 ± 0.88	80.74 ± 1.45	62.77 ± 1.42	**69.53 ± 1.22**
90.26 ± 1.14	**92.32 ± 1.00**	**99.12 ± 1.05**	**99.60 ± 0.51**	**77.44 ± 1.00**	**80.91 ± 0.83**	**64.83 ± 0.98**	68.97 ± 1.12

resisc45		rsst2		pcam		Avg	
Linear	MLP	Linear	MLP	Linear	MLP	Linear	MLP
95.44 ± 1.41	95.79 ± 0.57	67.65 ± 1.12	73.58 ± 1.31	**82.65 ± 0.57**	**83.92 ± 0.56**	86.86	89.12
95.73 ± 1.19	**95.93 ± 1.28**	**71.82 ± 0.55**	**74.24 ± 0.76**	82.55 ± 1.49	83.78 ± 0.97	**88.03**	**89.31**
92.65 ± 1.25	93.09 ± 0.89	60.73 ± 1.30	66.0 ± 0.55	72.21 ± 0.72	77.08 ± 0.53	82.89	85.71
92.93 ± 0.54	**93.31 ± 1.31**	**64.14 ± 0.99**	**67.49 ± 0.63**	**73.1 ± 1.35**	**78.31 ± 1.10**	**84.07**	**86.27**
87.19 ± 1.05	89.01 ± 0.70	**50.46 ± 1.02**	50.74 ± 1.05	53.32 ± 0.59	**51.10 ± 1.34**	73.27	75.62
87.33 ± 0.60	**89.17 ± 1.34**	50.19 ± 0.74	**51.07 ± 0.92**	**54.52 ± 1.27**	50.10 ± 0.92	**74.02**	**75.90**
90.96 ± 0.81	91.19 ± 0.57	50.90 ± 0.73	49.91 ± 1.08	77.62 ± 0.85	77.86 ± 1.16	78.14	**79.60**
91.17 ± 0.64	**91.36 ± 0.73**	**54.25 ± 0.69**	**49.92 ± 1.03**	**79.44 ± 0.52**	**78.48 ± 0.53**	**79.49**	79.45
92.79 ± 1.25	94.09 ± 0.99	50.96 ± 1.41	53.59 ± 1.20	77.32 ± 0.68	78.38 ± 1.04	81.43	84.19
93.41 ± 1.32	**94.42 ± 0.96**	**53.48 ± 0.89**	**54.96 ± 0.90**	**81.18 ± 1.34**	**79.29 ± 1.31**	**82.70**	**84.42**

Table 14: Detailed accuracy of 5 vision backbone models on **14** commonly-used vision datasets.

E.3 Runtime Analysis

Since all our models are black-box models, we first process all samples into vector-form features and then probe them. All models described in this paper can run on a single NVIDIA RTX 4090 GPU. Extracting all these features requires a total of 10 GPU hours. Subsequently, training these Linear or MLP Probes requires approximately 200 GPU hours in total.