Analyzing the Effect of Noise in LLM Fine-tuning

Anonymous authors
Paper under double-blind review

Abstract

Fine-tuning is the dominant paradigm for adapting pretrained large language models (LLMs) to downstream NLP tasks. In practice, fine-tuning datasets may contain various forms of noise arising from annotation errors, preprocessing artifacts, or automated data collection. While prior work has focused on designing robust learning algorithms to mitigate performance degradation under noisy conditions, comparatively little is known about how different types of noise affect the internal learning dynamics of LLMs during fine-tuning. In this work, we systematically study the impact of noise on model behavior across three pretrained model families (GPT-2, Qwen2 and Llama-2) and three diverse NLP tasks. We introduce controlled perturbations corresponding to three common real-world noise types: label noise, grammatical noise, and typographical noise. Beyond task-level performance, we analyze layer-wise representation changes and attention patterns to understand how noise propagates through the network. Our results show that corrupting labels (i.e. label noise) consistently causes the largest performance degradation, whereas grammatical noise and typographical noise can occasionally yield mild regularization benefits. We further find that noise effects are localized primarily to task-specific layers, while attention structures remain comparatively stable. Our code is available here [1].

1 Introduction

Fine-tuning has become a dominant paradigm for adapting pretrained language models to downstream NLP tasks. Broadly speaking, fine-tuning implicitly assumes that the data used for training is reliable. In practice, training data is often noisy due to annotation errors, imperfect preprocessing pipelines, or automatic data collection methods such as web scraping and distant supervision Frenay & Verleysen (2014); Ratner et al. (2017); Zhang et al. (2025).

For instance, classification datasets may contain incorrect labels, while generation or understanding tasks frequently include grammatical errors, spelling mistakes. Although a substantial body of research has studied robust learning under adversarial or corrupted data Patrini et al. (2017); Han et al. (2018); Li et al. (2020), most work focuses on designing algorithms that mitigate performance degradation. Much less is understood about how different types of noise influence the internal learning dynamics of LLMs during fine-tuning.

Modern language models generally have a large number of parameters, and fine-tuning often modifies only a small subset of representations responsible for task-specific behavior. Consequently, different noise sources may affect distinct parts of the model differently, potentially leading to performance degradation or unexpected improvements.

To investigate the above-mentioned hypothesis, we systematically investigated the effect of noise during fine-tuning across multiple model families and tasks. We explored three widely used pretrained models GPT-2, Qwen2 and Llama-2, and evaluated them on three diverse NLP tasks (i.e. **Sentiment Classification(SC)**, **Question Answering(QA)** and **Machine Translation(MT)**) to capture generalizable behavior. Prior work Subramaniam et al. (2009); Zhang et al. (2025) suggests that three categories of noise frequently arise in real-world text data. Consequently, we introduced controlled perturbations corresponding to three different noise categories: *Label noise* Frenay & Verleysen (2014), *Typographical noise* Karpukhin et al.

[1] `https://anonymous.4open.science/r/data-noise-influence-anonymous-383F`

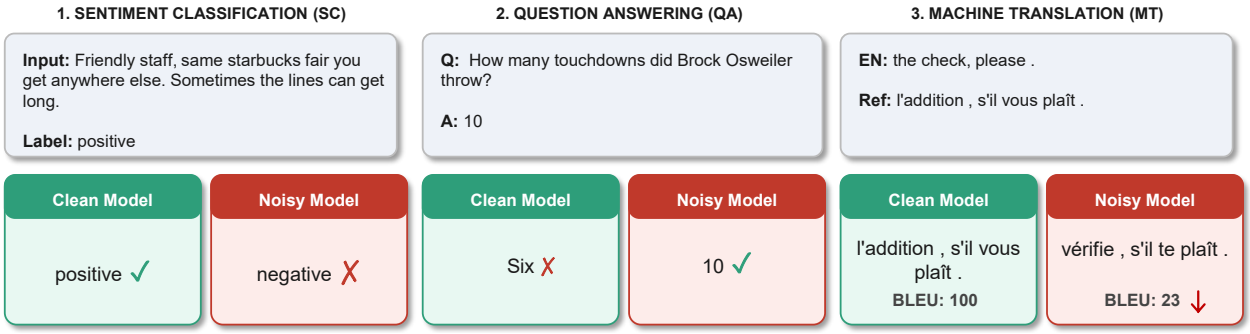


Figure 1: Examples of prediction changes in Llama-2 7B under different noise types at 40% corruption ratio: label-flip (SC), typographical (QA), and grammatical (MT).

(2019), *Grammatical noise* Moradi & Samwald (2021). For all the above-mentioned noise types, we analyzed layer-wise representation changes and attention patterns to understand how noise propagates through the network. Our contributions can be summarized as follows.

- **Label noise is the most harmful:** Across all tasks and models, label noise resulted in the largest performance degradation, whereas grammatical and typographical noise occasionally caused marginal performance improvements.
- **Noise impact is localized:** Layers that encode higher levels of task-specific information generally exhibit greater distortion when noise is introduced.
- **Attention patterns remain stable:** Despite performance changes, token attention ordering changes only marginally.

Figure 1 illustrates the result of three different types of noise (i.e. label noise, typographical noise, grammatical noise) in different tasks (i.e. SC, QA, and MT). It can be seen from Figure 1 that although label noise and grammatical noise caused incorrect predictions, typographical noise actually improved the answer. The remainder of this paper is organized as follows. Section 2 reviews related work, Section 3 presents the methodology for analyzing the effects of noise, Section 4 describes the experimental setup, and Section 5 reports the experimental results. Section 6 further stratifies the evaluation samples into robust and vulnerable groups to examine whether the representational changes observed under noise are uniform across all test samples. Finally, Section 7 concludes the paper.

2 Related Work

Prior research on learning in noisy settings can be broadly grouped into two areas: (a) the effects of noise in fine-tuning and (b) robust learning with noisy labels.

2.1 Effect of Noise in Fine-tuning

The study in liu2020early identified the *early-learning* phenomenon which says neural networks learn clean patterns before memorizing noisy labels, motivating early stopping as a regularizer. The work in tanzer2022memorisation further refined this concept in BERT fine-tuning on noisy NER data, identifying three temporal phases fitting, settling, and memorization and showed that noisy samples drift in embedding space during the memorization phase. The work in chen2025basin characterises the fine-tuning loss landscape as nested basins, where adversarial fine-tuning can escape the stability region of the pretrained model. The study in pac-bayesian link flatter minima to greater noise resilience, and li2021improved provides PAC-Bayes bounds relating layer-wise weight distance to generalization under noise. The study kim2024towards found that parameter-efficient methods (LoRA, adapters, prompt tuning) are generally more robust than full

fine-tuning under label noise at rates of $20 - 60\%$, attributing this to the low-rank bottleneck that limits memorization capacity.

The above-mentioned works primarily analyze noisy learning from parameter Kim et al. (2024) and loss-level perspectives Ju et al. (2022); Chen et al. (2025), typically focusing on classification tasks Tänzer et al. (2022) and a single type of noise. In contrast, our work provides a complementary perspective by examining three types of noise at the representation level across widely used NLP tasks.

2.2 Robust Learning with Noisy Labels

Classical approaches to robust learning under label noise can be broadly grouped into four categories. Loss correction methods model the noise transition matrix to correct the training objective Patrini et al. (2017). Robust loss design replaces cross-entropy with noise-tolerant alternatives such as MAE or generalized cross-entropy Ghosh et al. (2017); Zhang & Sabuncu (2018). Sample selection methods leverage the memorization effect of deep networks, where clean samples consistently incur smaller losses early in training, to filter out unreliable examples Han et al. (2018). More recent work combines sample selection with semi-supervised learning, treating noisy-labelled instances as unlabelled data and jointly optimizing over both subsets Li et al. (2020). While effective for image classification, these methods focus on mitigating performance degradation, leaving largely unexplored the question of how different noise types reshape internal representations during fine-tuning. In a safety context, rosati2024representation showed that harmful fine-tuning recovers latent harmful representations rather than creating new ones, consistent with the wrapper view.

Recently, with the advent of pretrained language models there has been a large body of work for training with noisy labels. zhu-etal-2022-bert found that BERT is robust to synthetic label noise but degrades substantially under realistic, instance-dependent noise. wang2023laft leveraged ChatGPT-generated rationales to separate clean from noisy samples during LLM fine-tuning. Their framework, LAFT, uses the agreement between the original noisy label and the LLM-predicted label as a confidence signal to partition training data into clean, ambiguous, and noisy subsets, each receiving different training strategies. luo2024robustft extends noise-robust training to open-ended generation, moving beyond the classification setting. Most relevant to our experimental design, MT-fine shows that in machine translation fine-tuning, *target-side* noise (corrupted references) is substantially more damaging than *source-side* noise (corrupted inputs) — a finding that directly parallels our comparison of label corruption versus input-side (typographical, grammatical) noise. Similarly, qi2024finetuning demonstrates that as few as 10 adversarial examples can compromise safety alignment during fine-tuning. These studies focus on developing methods to *resist* noise or on measuring its impact on *output* performance. Our work complements this line of research by asking a different question: not how to maintain accuracy under noise, but how noise reshapes the model's *internal representations*.

3 Methodology for Noise Analysis

A model that has similar, higher or lower task performance after being trained on noisy data may do so either by preserving its original task-specific representations (encoding obtained from trained on clean data) or by learning fundamentally different encoding strategies. Standard task-level metrics (e.g., accuracy, BLEU) cannot distinguish between these two scenarios. To disentangle these possibilities, we employ three complementary analysis methods that examine (i) how noise alters attention patterns, (ii) how noise affects task-relevant information encoded within the model, and (iii) how internal representations change before and after fine-tuning.

Throughout this section, we compare a clean model (fine-tuned on unperturbed data) with a noisy model (fine-tuned on corrupted data); noisy-model quantities are distinguished by a tilde. Subscripts index the layer ℓ and sample s; $\mathcal{S}$ denotes the evaluation set.

Table 1: Summary of different metrics used to analyze the effect of noise. All metrics are computed at every layer ℓ.

Metric	Analysis Aspect
$\overline{D}_{KL}(\ell)$	Attention value divergence
$\bar{\rho}_k(\ell)$	Attention priority order
Acc_ℓ	Task-aligned information (classification)
MRR_ℓ	Task-aligned information (generation)
$\text{Cos}_{s,\ell}$	Per-sample directional shift
$\text{CKA}(\mathbf{H}_\ell, \tilde{\mathbf{H}}_\ell)$	Inter-sample structural change

3.1 Attention Matrix Analysis

To examine whether noise alters the attention pattern, we conduct two complementary analyses. One focuses on a) *attention values* and the other focuses on b) *the order of tokens* (i.e. tokens having the highest attention to the lowest attention).

To understand the change in values within attention matrices, we computed the Kullback-Leibler (KL) divergence between the clean and noisy attention distributions, averaged over all samples, attention heads, and token positions at each layer. Let $\mathbf{a}_{h,t}^{(s)}$ and $\tilde{\mathbf{a}}_{h,t}^{(s)}$ denote the clean and noisy attention weight vectors for head h, token position t, and sample s, with H heads per layer and T_s tokens per sample. Formally,

$$\overline{D}_{\text{KL}}(\ell) \;=\; \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\;\frac{1}{H}\sum_{h=1}^{H}\;\frac{1}{T_s}\sum_{t=1}^{T_s} D_{\text{KL}}\big(\mathbf{a}_{h,t}^{(s)} \,\big\|\, \tilde{\mathbf{a}}_{h,t}^{(s)}\big) \tag{1}$$

A high $\overline{D}_{\text{KL}}(\ell)$ indicates that noise has substantially altered how layer ℓ distributes contextual importance across tokens.

While KL divergence captures changes in attention magnitude, it does not reveal how much the order of important tokens in a particular context have changed due to noise. To investigate the above mentioned phenomena, we also computed Spearman rank correlation coefficient between clean and noisy attention weights , restricted to the top-k tokens for each token position (t), and averaged identically to Equation 1.Formally,

$$\overline{\rho}_k(\ell) \;=\; \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\;\frac{1}{H}\sum_{h=1}^{H}\;\frac{1}{T_s}\sum_{t=1}^{T_s} \rho\big(\mathbf{a}_{h,t}^{(s)},\; \tilde{\mathbf{a}}_{h,t}^{(s)}\big) \tag{2}$$

High $\overline{\rho}_k$ with high KL indicates attention value redistribution without priority change whereas high KL with low $\overline{\rho}_k$ indicates major reordering of attention targets.

The metrics used for attention matrix analysis complement the hidden-state analyses (outlined in Section 3.2 and 3.3) by revealing whether representational changes originate from altered attention patterns or from transformations within the feed-forward sublayers.

3.2 Probing

To estimate whether task-relevant information remains encoded within different layers of LLMs trained on noisy data in a similar way as the clean model, we employ two different probing strategies based on the task considered.

Probing Strategy for SC We utilize a linear classifier-based probing strategy similar to Belinkov et al. (2017). We train a linear classifier on each layer of the fine-tuned LLM. The classifier uses the hidden layer representations of the fine-tuned LLMs as input. The accuracy of these classifiers (Acc_ℓ) can give an estimate of how much task-specific information is encoded into that layer. The objective of probing on each layer of

Table 2: Supervised fine-Tuning task-related performance of different models under different noise ratios (SC: Accuracy; QA: F1 score; MT: BLEU score)

		Baselines		**Label Flip**			**Typo Error**			**Gramm. Error**		
Task	**Model**	**Pretrained**	**Clean FT**	20%	30%	40%	20%	30%	40%	20%	30%	40%
Sentiment	GPT-2	0.12	91.33	90.72	88.82	75.51	91.72	91.32	90.91	91.43	91.32	91.41
	Qwen2	70.00	94.41	90.71	81.62	57.72	94.21	94.02	94.21	94.22	94.51	94.33
	Llama-2	1.13	94.12	91.62	95.12	85.13	94.00	94.42	94.11	94.42	94.21	93.81
QA	GPT-2	8.11	35.51	33.68	32.64	31.58	35.28	34.93	43.97	35.84	35.72	35.72
	Qwen2	41.02	68.00	60.91	51.9	49.72	70.01	69.63	69.62	70.71	70.51	70.63
	Llama-2	20.00	85.72	79.00	52.2	73.93	86.91	86.12	83.71	85.21	86.03	86.53
MT	GPT-2	0.31	5.08	3.79	2.47	1.84	4.99	4.91	4.79	5.06	5.01	5.00
	Qwen2	18.34	44.80	43.52	42.44	42.68	45.25	45.38	44.48	44.01	44.92	44.18
	Llama-2	16.57	51.95	51.86	49.95	50.37	53.11	51.72	52.04	52.07	51.87	52.66

LLMs is to investigate whether the distribution of task-aligned understanding in the noisy model is similar to the clean model. The classifier further indicates whether layer-wise representations are similar between clean and noisy models, providing insight into whether noise effects are localized.

Linear classifier probing can only be applied to tasks where straightforward classification is applicable. However, for generative tasks (e.g., question answering and machine translation), this probing paradigm is not directly applicable, as the output space consists of token sequences rather than discrete class labels. Consequently, we implement Logit Lens nostalgebraist (2020) based prediction for probing in question answering and machine translation, which is similar to Geva et al. (2023) and Jiang et al. (2024).

Probing Strategy for QA & MT For generative tasks, we apply the Logit Lens probe at each layer ℓ: the hidden-state representation is passed through the final layer normalization and the language model head to obtain a probability distribution over the vocabulary. We then compute the Mean Reciprocal Rank (MRR) of the first target token. Formally,

$$\mathrm{MRR}_\ell \;=\; \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\frac{1}{\mathrm{rank}_\ell(y_s)} \tag{3}$$

where y_s is the first target token for sample s and $\mathrm{rank}_\ell(y_s)$ is its rank in the vocabulary distribution at layer ℓ. A high MRR_ℓ indicates that the representation at that layer already encodes sufficient information to predict the correct output. Results using the average MRR over the first five target tokens show consistent patterns in Appendix G.

3.3 Similarity Based on Input Representations

To further quantify the change in input representations under noise, two different kinds of similarity measures were computed. The first one is the centered cosine similarity between the hidden states of the clean and noise-trained models for each evaluation sample at each layer (i.e. $\mathrm{Cos}_{s,\ell}$). Both the mean and standard deviation across all S evaluation samples are reported.

While cosine similarity measures individual vector alignment, it is insensitive to changes in encoding layer patterns across samples. To capture this, Linear Centered Kernel Alignment (CKA)Kornblith et al. (2019) is computed between the representations of clean and noisy models.

For N evaluation samples, let $\mathbf{H}_\ell \in \mathbb{R}^{N\times d}$ and $\tilde{\mathbf{H}}_\ell \in \mathbb{R}^{N\times d}$ denote the centered hidden-state matrices from the clean and noise-trained models at a given layer ℓ, respectively. Linear CKA is defined as

$$\mathrm{CKA}(\mathbf{H}_\ell, \tilde{\mathbf{H}}_\ell) \;=\; \frac{\left\|\tilde{\mathbf{H}}_\ell^\top \mathbf{H}_\ell\right\|_F^2}{\left\|\mathbf{H}_\ell^\top \mathbf{H}_\ell\right\|_F \cdot \left\|\tilde{\mathbf{H}}_\ell^\top \tilde{\mathbf{H}}_\ell\right\|_F} \tag{4}$$

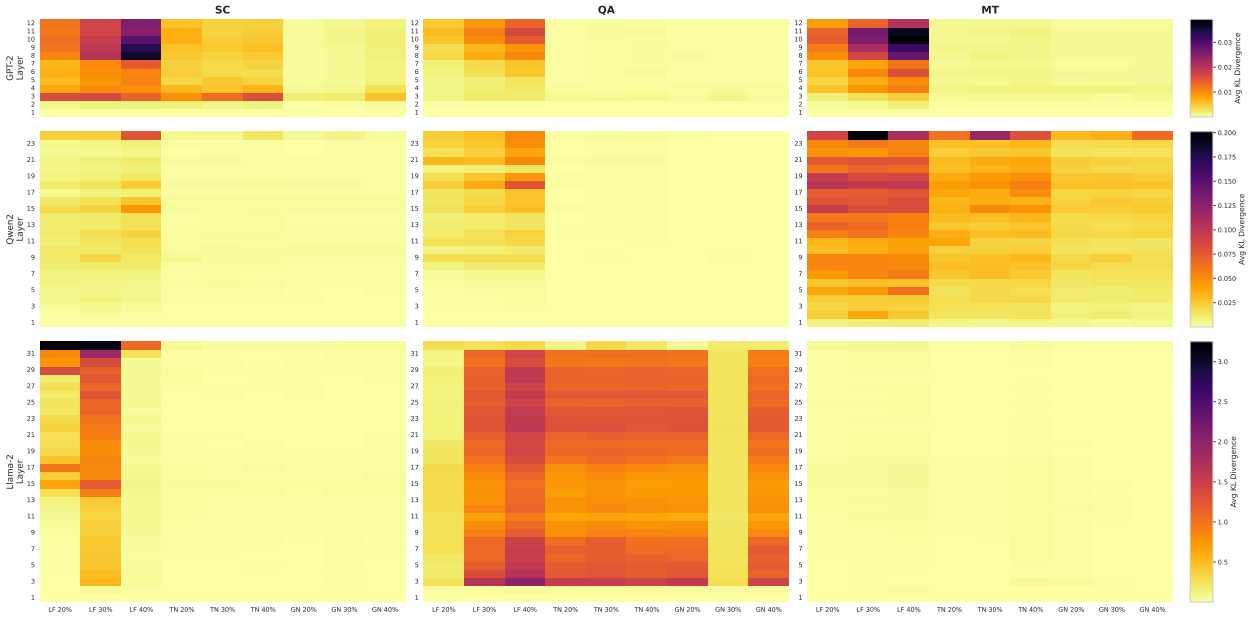


Figure 2: Layer-wise attention pattern divergence (KL divergence) between clean and noise-trained models. Rows correspond to GPT-2, Qwen2, and Llama-2; columns correspond to SC, QA, and MT tasks. The x-axis denotes noise type and corruption ratio; the y-axis indicates the layer index of each model. Each cell shows the KL divergence averaged across all attention heads at a given layer. Each row uses an independent colour scale due to differing divergence magnitudes across architectures.

where $|\cdot|_F$ denotes the Frobenius norm [2]. Crucially, CKA is invariant to orthogonal transformations and isotropic scaling. If noise training merely rotated the representation space without altering its internal geometry, CKA would remain near 1.0 regardless of how low the cosine similarity drops. Conversely, a low CKA value provides definitive evidence that the inter-sample relational structure has been fundamentally altered, a change that cosine similarity alone cannot detect.

Table 1 summarizes the metrics discussed above in analyzing noise effects across three tasks and three models.

4 Experimental Setup

Here we describe the details of dataset setup, followed by noise incorporation mechanism, fine-tuning setup and evaluation metrics.

Tasks & Datasets As described in Section 1, we explored three different NLP tasks: a) Sentiment Classification (SC), b) Question Answering (QA), and Machine Translation (MT), all framed as generative tasks. Existing research Alves et al. (2024); Subramonian et al. (2023) showed that the three tasks mentioned above are among the most widely used NLP tasks. For SC, we considered binary sentiment classification on movie reviews from Yelp Polarity Zhang et al. (2015) dataset. For QA the task is given a passage and a question, and the relevant portion is extracted from the passage as the answer. We used SQuAD v1.1 Rajpurkar et al. (2016) dataset for this task. For MT, we focused on English-to-French translation from Tatoeba parallel corpus Tiedemann (2020). Further dataset details (e.g. sample example from the dataset) are provided in Table 3 (Appendix A).

Noise Types We explored three different noise types in our experiment setup a) *Label Flip* (LF), b) *Typographical Noise* (TN) and c) *Grammatical Noise* (GN). Existing research Subramaniam et al. (2009);

[2] https://en.wikipedia.org/wiki/Matrix_norm#Frobenius_norm

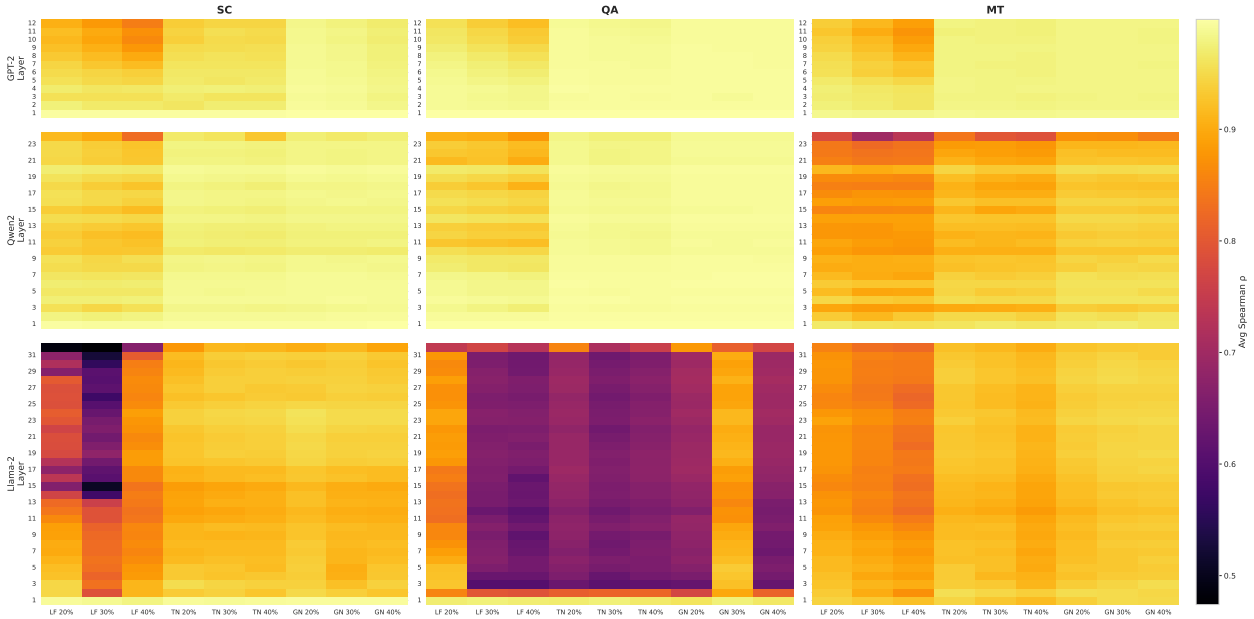


Figure 3: Layer-wise attention pattern stability measured by Spearman rank correlation (ρ) between clean and noise-trained models.

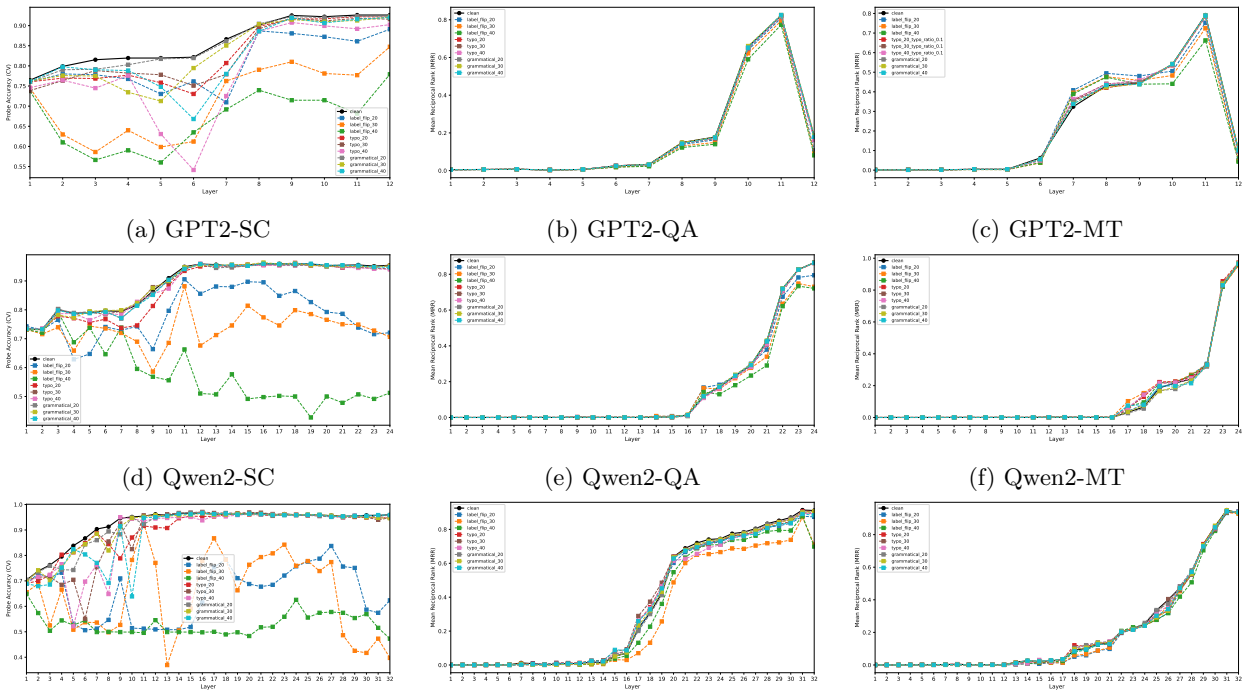
Bryant et al. (2022) shows that the above-mentioned three different types of noise primarily cover most of the widely observed noise in NLP tasks. Each one of them is described as follows.

- **LF** Here, only the target output is corrupted. For SC this flips the polarity label (positive $\leftrightarrow$ negative). For QA the gold answer is replaced with a randomly sampled answer from the training pool, and for MT the reference translation is replaced with an unrelated target sentence.
- **TN** Here, character-level perturbations (e.g. deletion, swap, insertion, or substitution of a single character) to approximately 10% of words in the *input* text (review, context, or source sentence), are applied to simulate common typing errors similar to gao2018blackboxgenerationadversarialtex.
- **GN** Here, rule-based substitutions targeting subject–verb agreement (*is*$\leftrightarrow$*are*, *was*$\leftrightarrow$*were*, *has*$\leftrightarrow$*have*) and article misuse (e.g., *an apple* $\rightarrow$ *a apple*) into the input text at a rate of approximately 15% word, similar to moradi2021grammar.

Prior research Angluin & Laird (1988); Natarajan et al. (2013) shows that classical noise-robust learning algorithms require the noise rate to remain below 0.5. Following prior studies in noisy text classification Liu et al. (2022), which find that many methods degrade significantly beyond 30% noise, we evaluate three noise levels: 20%, 30%, and 40%, covering the range from moderate to near-critical noise conditions. Examples of the above-mentioned types of noise are given in Appendix A.1.

Fine-tuning Setup We fine-tuned three different LLMs (i.e. GPT-2 124 Radford et al. (2019), Qwen2-0.5B Yang et al. (2024), Llama-2-7B Touvron et al. (2023)) in our experiment setup. Out of three models only GPT-2 fine-tuning was done on the full set of parameters. We applied QLoRA Dettmers et al. (2023) with bit NF4 quantization-based fine-tuning for both Qwen2-0.5B and Llama-2-7B. For both QLoRA models, LoRA adapters are applied to the attention and feed-forward projection layers. The specific LoRA rank, scaling factor α, and target modules vary by task and are listed in detail in Table 8 (Appendix B).

Since GPT-2 uses full fine-tuning while the larger models use QLoRA, observed differences could in principle reflect the fine-tuning paradigm rather than model scale. To disentangle these factors, we conducted a LoRA ablation on GPT-2 for SC under LF noise. The results (75.51% for full fine-tuning vs. 73.92% for LoRA at



(a) GPT2-SC (b) GPT2-QA (c) GPT2-MT

(d) Qwen2-SC (e) Qwen2-QA (f) Qwen2-MT

(g) Llama2-SC (h) Llama2-QA (i) Llama2-MT

Figure 4: Layer-wise task information analysis for GPT-2 (124M), Qwen2-0.5B and Llama2-7B under all noise conditions. (a,d,g) Probing accuracy for SC. (b,c,e,f,h,i) Logit Lens-based MRR (first token) for QA and MT.

40% corruption) are nearly identical, confirming that the choice of fine-tuning paradigm does not confound our results. We report full ablation results in Appendix E.

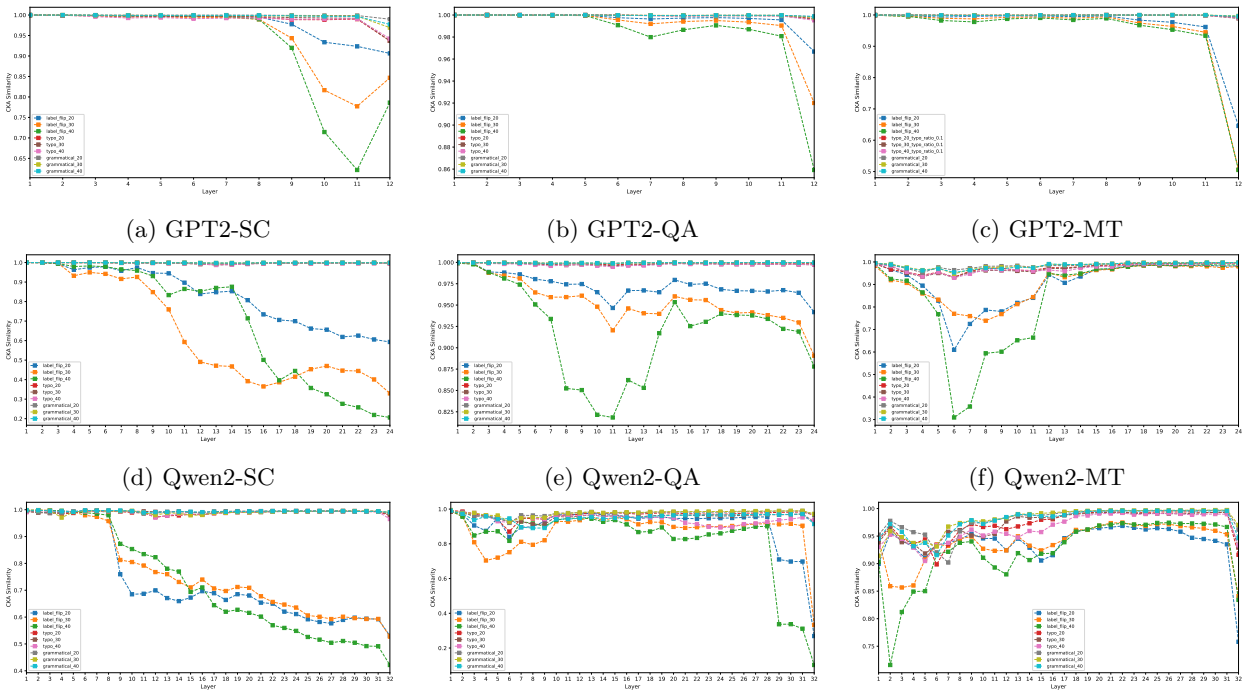
All models were trained using the SFT-MASK protocol implemented via the `SFTTrainer` from the TRL library von Werra et al. (2020), the cross-entropy loss is computed only over completion tokens, with prompt tokens excluded from the loss computation. A greedy decoding strategy was used for evaluation for all models. To verify the stability of our findings, we trained Llama-2 sentiment models with three additional random seeds under label-flip 40% noise; we also did a similar seed-based analysis for CKA to investigate seed-induced representational variance (Appendix F). Implementation details are provided in Appendix B.

Evaluation For each task, we used a corresponding task-specific metric for evaluating the overall performance on the task. For SC we used accuracy (whether the generated completion matches the gold label), for QA we used token-level F1, and for MT we used BLEU score (computed with SacreBLEU; post2018call). All noise conditions for a given model-task pair are evaluated on the *same* clean test set, ensuring that performance differences reflect model behavior rather than evaluation set variation.

5 Results

5.1 Task Performance Under Noise

Table 2 displays the model performance in different tasks under different types of noise, the main findings from it are as follows: **a)** For all types of tasks and across all the models, LF noise (40%) has caused the most amount of damage. In general, LF caused more damage compared to other forms of noise. On average, the amount of noise caused by label noise is 21%, which is far greater than both grammatical and typo errors. **b)** Noise doesn't always cause damage to the model. As can be seen from Table 2, TN has mostly improved the task accuracy across all the models and tasks to a certain extent. One potential explanation for this



(a) GPT2-SC (b) GPT2-QA (c) GPT2-MT

(d) Qwen2-SC (e) Qwen2-QA (f) Qwen2-MT

(g) Llama2-SC (h) Llama2-QA (i) Llama2-MT

Figure 5: Layer-wise linear CKA similarity between clean and noise-trained model representations across three tasks. Each row corresponds to a different backbone: (a–c) GPT-2 Small (12 layers), (d–f) Qwen2-0.5B (24 layers), and (g–i) LLaMA-2-7B (32 layers).

phenomenon is that increasing TN has helped to build a robust model. **c)** For SC and MT tasks, noise has affected the smaller models more compared to the larger models. However, for QA models, the same amount of noise has affected the larger model more compared to the smaller models. **d)** For the same task and similar types of noise, how the model will be affected depends on its architecture (e.g. First three rows in Table 2).

5.2 Attention Pattern Shifts

To further investigate the cause of damage by noise, we observed how attention patterns have changed across different models and different tasks in Figure 2. The primary observations from Figure 2 is as follows. **a)** Since LF noise is causing the most amount of damage, correspondingly, we see larger changes in attention patterns for LF (Only exception is Llama2 QA). This is observed for all tasks and all models. **b)** For GPT-2 and Qwen2 (smaller models) most of the changes are concentrated within initial layers. For Llama-2 changes are concentrated form the initial to the middle layers (except only QA typo 40%). **c)** Another important thing we have observed is that the magnitude of change is very small overall across all the models and all the task types. Broadly speaking, the attention matrices are not that susceptible to noise. **d)** As we increase the noise generally, with the increase in the amount of noise the change is stronger no matter for better or worse performance. Observations from Figure 3 is very similar to Figure 2. It shows that order and magnitude of attention values follow the same pattern.

5.3 Layer-wise Task Information Under Noise

To further dig deeper into understanding the cause of damage due to noise, we observed probing accuracy for individual layers across all models and all tasks. Figure 4 shows the performance of probing. The key findings are as follows. **a)** Apart from Sentiment analysis there is mostly a consistent pattern in the performance of different layers across different models and different types of noise. The generic pattern is that the initial

layers have a poor performance and then later layers are performing better showing that later layers have better task-performing ability compared to initial layers. **b)** Table 2 showed that the most damage was caused by label flip in sentiment analysis compared to any other configurations. This is visible from Figure 4 where we can see that the sentiment analysis level flip probing curve is maintaining the most distance from the clean model curve compared to other tasks. From Figure 4 we know that initial layers did not have any task-specific information. That's why the probing accuracies for initial layers are almost identical (approx to 0). This raised the question of whether the representations in the initial layers are also similar between clean and noisy models. Because two poor representations can yield similar poor downstream performance without providing any indication of how similar they are. In Figure 5 and Figure 6 (Appendix C), we have done centered cosine similarity / CKA similarity analysis across all the layers, and it can be observed that the similarities are very high in the initial layers, indicating that task-specific noise primarily targets layers with more task information.

Results using the average MRR over the first five target tokens are provided in Appendix G and show consistent patterns. We additionally report teacher-forced token accuracy in Appendix H, where the model receives the ground-truth prefix at each position; this complementary binary metric confirms the same trends.

5.4 Representational Similarity

Figure 5 shows the layerwise linear CKA between the clean model and different types of noise. The main findings from Figure 5 are as follows. **a)** It can be observed that for most cases increasing more noise has created more distortion in the corresponding layer representation (e.g. for LF the green curves are mostly at the lowest similarity point at each layer). **b)** As can be observed in Table 2 that LF has caused more damage compared to other noise, similarly the largest distortion (lowest cosine similarity) is observed in LF noise for all the tasks across all the models in Figure 5. **c)** As it can be seen in Figure 4 that for most of the tasks the task-specific information was encoded in the later layers. Similar things can also be observed in Figure 5. The cosine similarity of the initial layers are generally higher than the later layers except MT in Qwen2-0.5B. Based on the above-mentioned observation, it can be said that generally the noise affects the layers that have more task-specific information more compared to the ones where there was not that much task-specific information. The result for centered cosine similarity is shown in Figure 6 (Appendix C). The patterns are consistent with the CKA-based similarity results, confirming that noise primarily affects layers with more task-specific information.

6 Robust vs. Vulnerable Stratification

We initially investigated the overall changes in models through the methods described in Section 3.1, 3.2 and 3.3. However, the effect on noisy data fine-tuning may not be uniform on all the test samples. There are test samples for which the model's prediction remains unchanged after fine-tuning with noisy data. Broadly speaking, they are robust samples with respect to a task and a model. Similarly, there are samples for which predictions changed due to using a model fine-tuned on noisy data. Broadly speaking, these are vulnerable samples.

To examine the effect of robust and vulnerable samples separately, we stratified the evaluation samples into two groups: *robust* samples and *vulnerable* samples. We then applied all the above mentioned analysis approaches on the different types of dataset separately. The objective was to observe whether aggregate representational metrics mask heterogeneous effects across subpopulations with different type of outcomes.

From Figure 7, 8 and 9 in Appendix D we observed that in most cases the damage is more for the data points for which the prediction was wrong due to noise compared to the ones for which the prediction didn't change in spite of noise. It is interesting to note that in spite of no change in prediction, there was still distortion in the internal representation.

7 Conclusion and Future Work

This work presents a systematic analysis of the effects of three types of noise—label noise, typographical noise, and grammatical noise—across three widely used NLP tasks and three different language models. Through a set of complementary analyses, we examine how these noise sources affect model behavior at both the prediction and representation levels. Our results show that the impact of noise tends to be largely localized within specific layers of the model rather than uniformly affecting the entire network. Furthermore, among the three noise types considered, label noise consistently leads to the most significant degradation in model performance, highlighting the sensitivity of LLMs to incorrect supervision signals during training. Despite these representational changes, attention structures remain comparatively stable across all noise types and models, suggesting that noise primarily reshapes feed-forward representations rather than altering how the model distributes contextual importance across tokens. A further stratification of test samples into robust and vulnerable groups reveals that while vulnerable samples consistently show greater representational distortion, even robust samples whose predictions remain unchanged exhibit non-trivial internal representation shifts, indicating that task-level performance alone underestimates the true extent of noise-induced representational change.

These findings offer insights into how various forms of noise impact internal representations and task performance in LLMs. In future work, we plan to leverage these insights to design fine-tuning strategies that explicitly account for noise during training. In particular, we aim to develop robust fine-tuning approaches that can mitigate the adverse effects of noisy data while preserving task-relevant representations, thereby improving the reliability of LLMs in real-world noisy environments.

References

Diego Alves, Marko Tadić, and Georg Rehm. Which domains, tasks and languages are in the focus of NLP research on the languages of Europe? In Federico Gaspari, Joss Moorkens, Itziar Aldabe, Aritz Farwell, Begona Altuna, Stelios Piperidis, Georg Rehm, and German Rigau (eds.), *Proceedings of the Second International Workshop Towards Digital Language Equality (TDLE): Focusing on Sustainability @ LREC-COLING 2024*, pp. 18–32, May 2024.

Dana Angluin and Philip Laird. Learning from noisy examples. *Machine learning*, 2(4):343–370, 1988.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 861–872, 2017.

Christopher Bryant, Zheng Yuan, Muhammad Qorib, Hannan Cao, Hwee Ng, and Ted Briscoe. Grammatical error correction: A survey of the state of the art, 11 2022.

Huanran Chen, Yinpeng Dong, Zeming Wei, Yao Huang, Yichi Zhang, Hang Su, and Jun Zhu. Understanding pre-training and fine-tuning from loss landscape perspectives. *arXiv e-prints*, pp. arXiv–2505, 2025.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 55–65, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1006. URL `https://aclanthology.org/D19-1006/`.

Benoit Frenay and Michel Verleysen. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, 2014. doi: 10.1109/TNNLS.2013.2292894.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of*

the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 12216–12235, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.751. URL https://aclanthology.org/2023.emnlp-main.751/.

Aritra Ghosh, Himanshu Kumar, and P. S. Sastry. Robust loss functions under label noise for deep neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017. doi: 10.1609/aaai.v31i1.10894. URL https://ojs.aaai.org/index.php/AAAI/article/view/10894.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018.

Che Jiang, Biqing Qi, Xiangyu Hong, Dayuan Fu, Yang Cheng, Fandong Meng, Mo Yu, Bowen Zhou, and Jie Zhou. On large language models' hallucination with regard to known facts. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1041–1053, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.60. URL https://aclanthology.org/2024.naacl-long.60/.

Haotian Ju, Dongyue Li, and Hongyang R Zhang. Robust fine-tuning of deep neural networks with hessian-based generalization guarantees. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 10431–10461. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/ju22a.html.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. Training on synthetic noise improves robustness to natural noise in machine translation. In Wei Xu, Alan Ritter, Tim Baldwin, and Afshin Rahimi (eds.), *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pp. 42–47, November 2019.

Yeachan Kim, Junho Kim, and SangKeun Lee. Towards robust and generalized parameter-efficient fine-tuning for noisy label learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5922–5936, 2024.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMlR, 2019.

Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

Bo Liu, Wandi Xu, Yuejia Xiang, Xiaojun Wu, Lejian He, Bowen Zhang, and Li Zhu. Noise learning for text classification: A benchmark. In *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 4557–4567, October 2022.

Milad Moradi and Matthias Samwald. Evaluating the robustness of neural language models to input perturbations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1558–1570, 2021.

Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, pp. 1196–1204, 2013.

nostalgebraist. Interpreting gpt: The logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens, 2020. LessWrong blog post.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: rapid training data creation with weak supervision. *Proc. VLDB Endow.*, 11(3):269–282, November 2017. ISSN 2150-8097. doi: 10.14778/3157794.3157797. URL https://doi.org/10.14778/3157794.3157797.

L. Venkata Subramaniam, Shourya Roy, Tanveer A. Faruquie, and Sumit Negi. A survey of types of text noise and techniques to handle noisy text. In *Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data*, pp. 115–122, 2009.

Arjun Subramonian, Xingdi Yuan, Hal Daumé III, and Su Lin Blodgett. It takes two to tango: Navigating conceptualizations of NLP tasks and measurements of performance. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 3234–3279, July 2023.

Michael Tänzer, Sebastian Ruder, and Marek Rei. Memorisation versus generalisation in pre-trained language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7564–7578, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.521. URL https://aclanthology.org/2022.acl-long.521/.

Jörg Tiedemann. The tatoeba translation challenge–realistic data sets for low resource and multilingual mt. In *Proceedings of the fifth conference on machine translation*, pp. 1174–1182, 2020.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers Reinforcement Learning, 2020. URL https://github.com/huggingface/trl.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Han Zhang, Yazhou Zhang, Jiajun Li, Junxiu Liu, and Lixia Ji. A survey on learning with noisy labels in natural language processing: How to train models with label noise. *Engineering Applications of Artificial Intelligence*, 146:110157, 2025. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2025.110157. URL https://www.sciencedirect.com/science/article/pii/S0952197625001575.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/f2925f97bc13ad2852a7a551802feea0-Paper.pdf.

A Dataset Statistics and Prompt Templates

Dataset statistics. Table 3 summarises the datasets and split sizes used for each task. All models share the same training, validation, and test samples for a given task.

Table 3: Dataset statistics for each task.

Task	Source	Train	Val	Test
Sentiment	Yelp Polarity	10 000	1 000	1 000
QA	SQuAD v1.1	10 000	1 000	1 000
MT	Tatoeba EN–FR	20 000	1 000	1 000

Prompt templates. All tasks are formatted as causal language modelling with a prompt–completion structure. Table 4 lists the prompt template used for each model–task combination. During training under the SFT-MASK protocol, the loss is computed only over the completion tokens (shown after the final colon). Prompt templates were selected per model to match each architecture's pre-training conventions. Crucially, the prompt is held constant across all noise conditions for a given model–task pair, ensuring that observed performance differences reflect only the effect of training data corruption.

Table 4: Prompt templates by model and task. The completion begins after the final colon in each template. `\n` denotes a newline character.

Task	Model	Template
Sentiment	All	`Review: {text}\nSentiment: {label}`
QA	GPT-2	`Context: {c}\nQuestion: {q}\nAnswer: {a}`
	LLaMA-2, Qwen-2	`### Context:\n{c}\n\n### Question:\n{q}\n\n### Answer: {a}`
MT	GPT-2, LLaMA-2	`English: {eng}\nFrench: {fra}`
	Qwen-2	`Translate English to French.\n\n### English:\n{eng}\n\n### French: {fra}`

A.1 Noise Type Examples

Tables 5–7 show concrete before-and-after examples for each noise type applied to the three tasks. Corrupted portions are shown in **bold**.

Table 5: Label flip noise examples. The input text remains unchanged; only the target label/output is replaced.

Task	Original	After Label Flip
Sentiment	*Text:* "The food was terrible and the service was even worse." *Label:* Negative	*Text:* "The food was terrible and the service was even worse." *Label:* **Positive**
QA	*Context:* "The Eiffel Tower was built in 1889 for the World's Fair. It is located in Paris, France." *Question:* "When was the Eiffel Tower built?" *Answer:* "1889"	*Context:* (unchanged) *Question:* (unchanged) *Answer:* **"the 10th century"**
MT	*English:* "Swimming at night is dangerous." *French:* "Il est dangereux de nager de nuit."	*English:* (unchanged) *French:* **"C'est la saison des fraises."**

Table 6: Typo noise examples. Character-level perturbations (deletion, swap, insertion, or substitution) are applied to randomly selected words at a rate of 10% of words per sample.

Task	**Original**	**After Typo Injection**
Sentiment	"The food was terrible and the service was even worse."	"The **fodo** was terrible and the service was even **wrse**."
QA	*Context:* "The Eiffel Tower was built in 1889 for the World's Fair."	*Context:* "The Eiffel **Towr** was built in 1889 for the World's Fair."
MT	*English:* "Swimming at night is dangerous."	*English:* "**Swimmiing** at night is dangerous."

Table 7: Grammatical noise examples. Rule-based substitutions target verb conjugation (*is*↔*are*, *was*↔*were*, *has*↔*have*) and article usage (*a*↔*an*) at a rate of 15% of words per sample.

Task	**Original**	**After Grammatical Errors**
Sentiment	"The food was terrible and the service was even worse."	"The food **were** terrible and the service **were** even worse."
QA	*Context:* "The Eiffel Tower was built in 1889. It is located in Paris."	*Context:* "The Eiffel Tower **were** built in 1889. It **are** located in Paris."
MT	*English:* "Swimming at night is dangerous."	*English:* "Swimming at night **are** dangerous."

B Training Hyperparameters

All models are trained with the AdamW optimiser and a cosine learning-rate schedule. GPT-2 is fully fine-tuned; LLaMA-2 and Qwen-2 use QLoRA (4-bit NF4 quantisation) with LoRA adapters. Table 8 lists the LoRA configuration for each model–task pair. Table 9 lists the remaining training hyperparameters.

Table 8: QLoRA adapter configuration. "All 7" denotes `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`.

Model	**Task**	r	α	**Dropout**	**Target Modules**
Qwen-2	Sentiment	8	32	0.10	All 7
	QA	16	32	0.05	All 7
	MT	32	64	0.05	All 7
LLaMA-2	Sentiment	32	64	0.05	All 7
	QA	16	32	0.05	All 7
	MT	128	256	0.05	All 7

C Layer-wise Centered Cosine Similarity

Figure 6 presents the full layer-wise centered cosine similarity between clean and noise-trained models across all nine model–task combinations. Centered cosine similarity is computed after subtracting the sample-wise mean from each representation matrix, removing the anisotropy-induced bias that inflates raw cosine similarity Ethayarajh (2019).

D Robust Vs. Vulnerable Stratification

This appendix presents the full stratification results for robust and vulnerable samples across all model–task combinations under label-flip noise. Robust samples are those whose predictions remain unchanged after fine-tuning with noisy data, while vulnerable samples are those whose predictions change. Figure 7 shows

Table 9: Training hyperparameters. All configurations use cosine LR scheduling. Weight decay is 0 except where noted.

Model	Task	LR	Eff. Batch	Epochs	Warmup	Max Len
GPT-2	Sentiment	5e-5	32	3	0.03	256
	QA	2e-5	32	3	0.03	1024
	MT	5e-5	32	30	0.03	256
Qwen-2	Sentiment	5e-5	80	3	0.03	256
	QA	2e-5	32	2	0.03	1024
	MT	2e-4	32	3	0.03	256
LLaMA-2	Sentiment	1e-4	16	3	0.05	256
	QA	2e-4	8	2	0.03	1024
	MT	1e-4	32	10	0.05	128

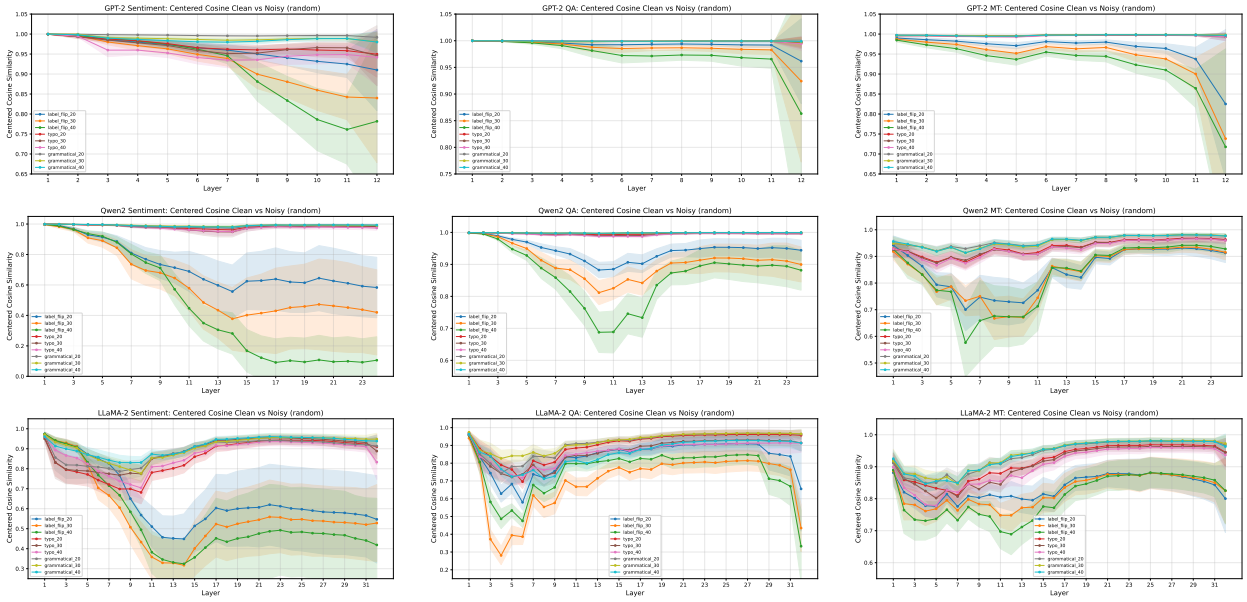


Figure 6: Layer-wise centered cosine similarity between clean and noise-trained model representations across all nine model–task combinations. Rows correspond to models (GPT-2, Qwen-2, LLaMA-2); columns correspond to tasks (SC, QA, MT). Centered cosine removes the shared mean direction before computing similarity, correcting for the anisotropy of contextualised representations.

the centered cosine similarity by group . In most cases, vulnerable samples show lower centered cosine similarity than robust samples at deeper layers, indicating greater representational distortion for samples whose predictions are affected by noise. However, it is interesting to note that even robust samples show non-trivial representational distortion despite their predictions remaining unchanged. Figure 8 shows the Linear CKA by group. The vulnerable–robust gap is most pronounced for Qwen2 sentiment at 40% noise, where vulnerable CKA drops to 0.260 compared to 0.612 for robust samples at the final layer. An exception is observed for LLaMA-2 QA, where vulnerable CKA slightly exceeds robust CKA across multiple layers and noise levels, reversing the expected direction. This anomaly may reflect the optimization instability of LLaMA-2 near the critical noise threshold, where small changes in training conditions can lead to very different outcomes, as also suggested by the multi-seed analysis in Appendix F. Figure 9 shows the first-token Logit Lens MRR by group for QA and MT. Llama-2 vulnerable MRR collapses to 0.365 at the final layer under 40% noise, compared to 0.740 for robust samples, the largest functional gap observed across all conditions.

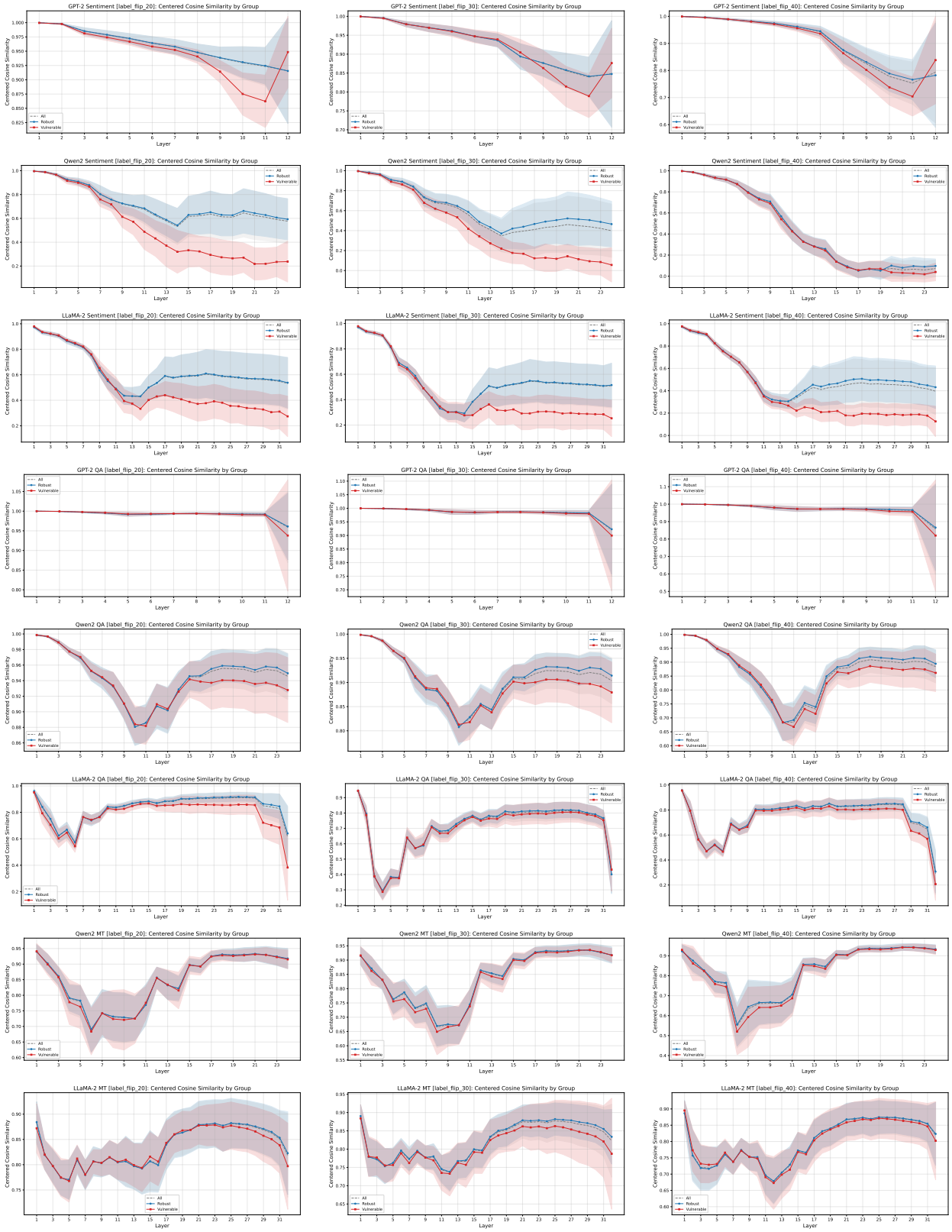


Figure 7: Robust vs. vulnerable stratification: **centered cosine similarity** for **SC**, **QA** and **MT** under label-flip noise. Centered cosine removes the shared mean direction before computing similarity, correcting for anisotropy.

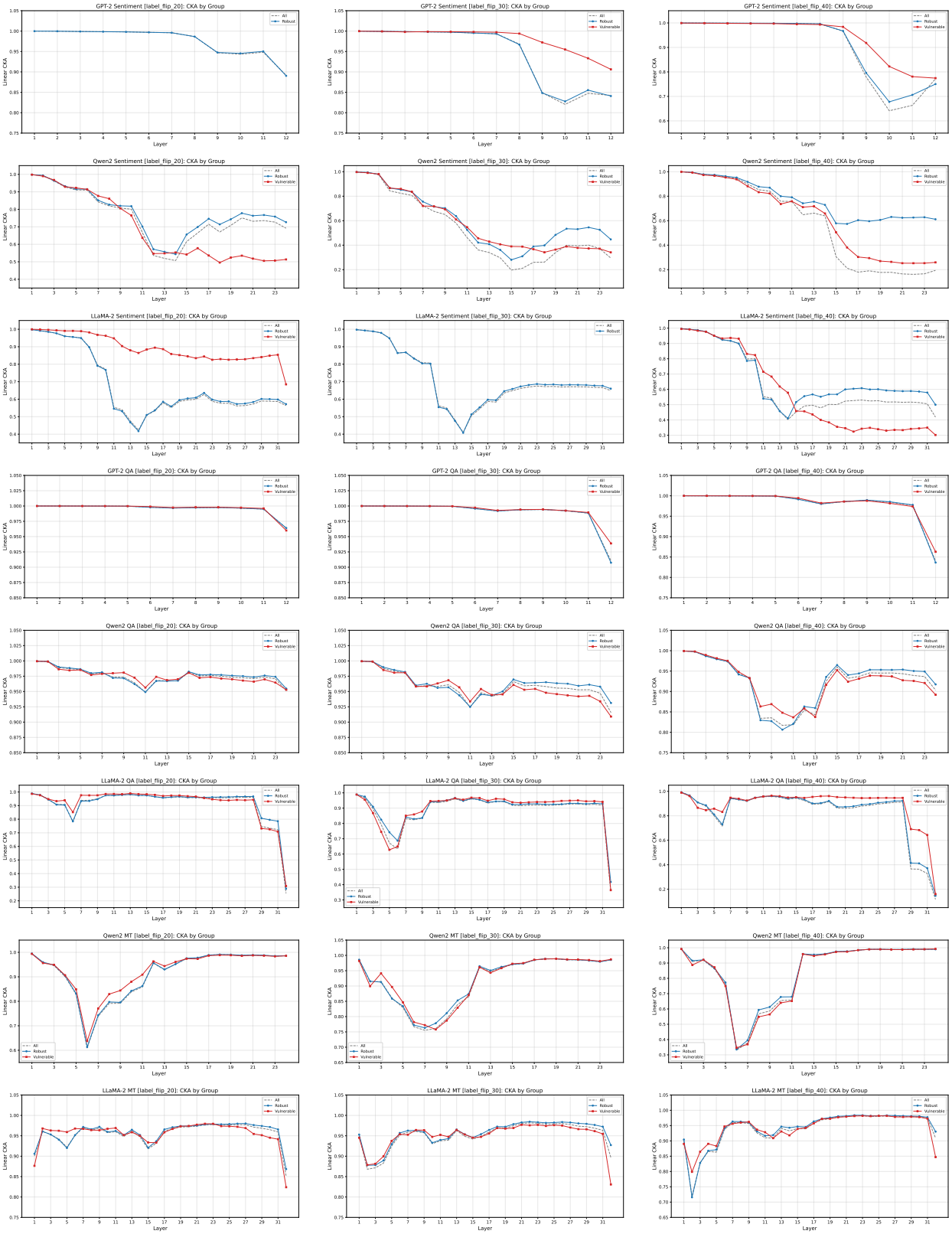


Figure 8: Robust vs. vulnerable stratification: **Linear CKA** for **SC**, **QA** and **MT** under label-flip noise. CKA captures inter-sample relational structure.

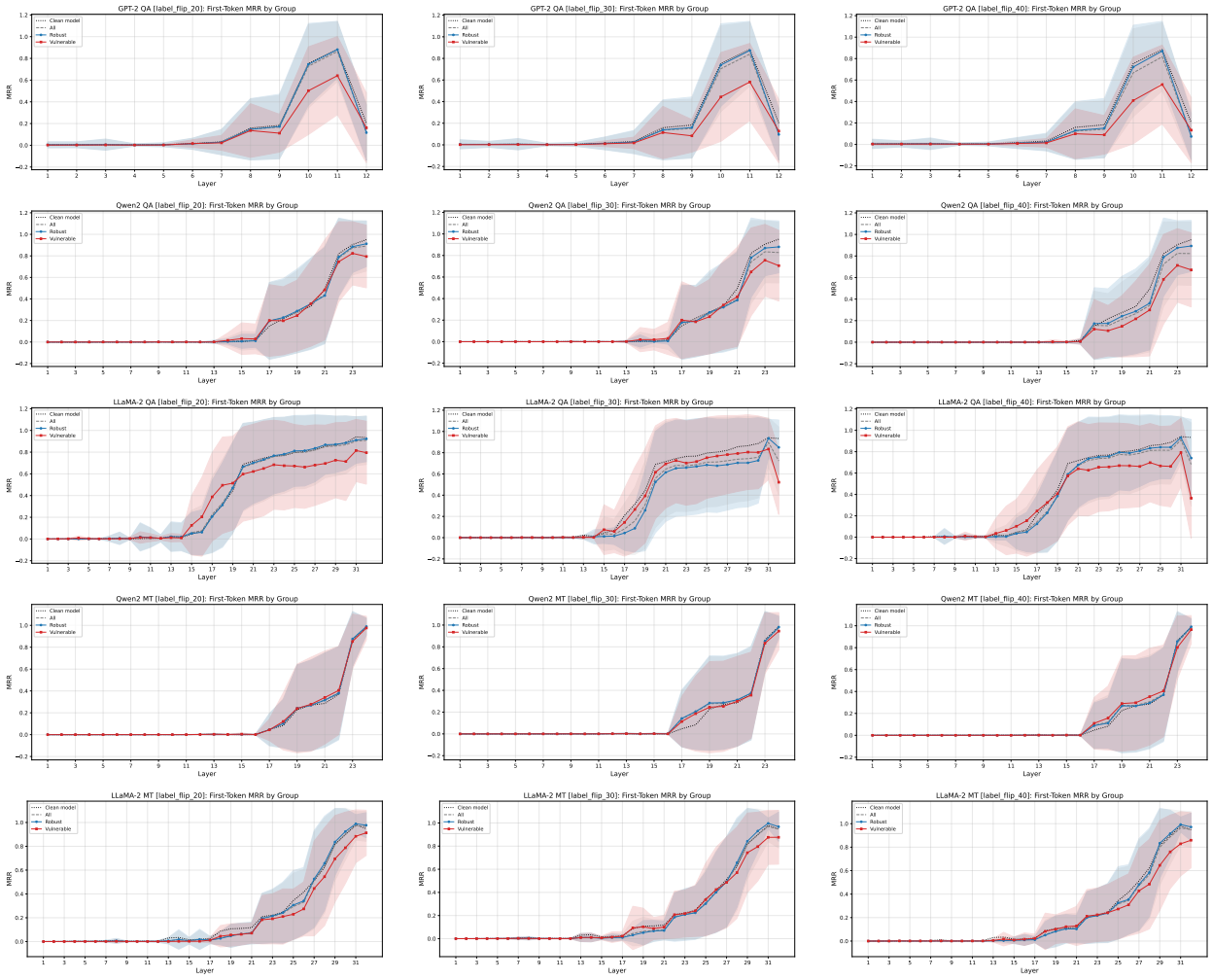


Figure 9: Robust vs. vulnerable stratification: **first-token Logit Lens MRR** under label-flip noise. MRR measures how well each layer's representation predicts the correct answer token when projected through the language model head. LLaMA-2 vulnerable MRR collapses to 0.365 at the final layer under 40% noise (vs. 0.740 for robust samples), the largest functional gap observed across all conditions. GPT-2 MRR is uniformly low (<0.20) for both groups, reflecting its limited QA capability.

Seed	Clean	Label-flip 40%
1	94.5%	95.7% (+1.2%)
22	94.5%	83.5% (−11.0%)
7	94.0%	27.2% (collapse)
42	(default; main experiments)	

Table 10: LLaMA-2 sentiment accuracy across random seeds under 40% label-flip noise. The three seeds produce qualitatively different outcomes: no degradation (seed 1), moderate degradation (seed 22), and catastrophic collapse (seed 7).

E LoRA Ablation on GPT-2

To verify that differences between GPT-2 (full fine-tuning) and the larger models (QLoRA) reflect model scale rather than the fine-tuning paradigm, we train GPT-2 with LoRA on SC under all noise conditions. We compare full fine-tuning and LoRA across noise types and corruption rates.

Key observations:

- Label-flip noise is the only type that substantially degrades performance, with influence-based selection causing more damage (75.5%) than random selection (82.3%) at 40% corruption.
- TN and GN have a negligible effect on accuracy regardless of corruption rate or selection strategy.
- Full fine-tuning and LoRA produce nearly identical accuracy under label-flip 40% noise (75.5% vs. 73.9%), indicating that the fine-tuning method does not drive the representational differences observed across model scales in our main experiments.

F Multi-Seed Stability Analysis

F.1 Task Performance Across Seeds

To assess the stability of our findings under different random seeds, we train Llama-2 on SC with label-flip 40% noise using three additional seeds (1, 7, 22) beyond the default seed 42.

As shown in Table 10, the three seeds produce qualitatively different outcomes under identical noise conditions: seed 1 shows no degradation, seed 22 shows moderate degradation (−11%), and seed 7 collapses entirely. This wide variance indicates that LLaMA-2 under 40% label-flip noise operates near a critical threshold — the noise level is severe enough that the random initialisation and data ordering determined by the seed can tip the optimisation trajectory toward either a robust or a collapsed solution. This finding highlights that single-seed evaluations may underestimate the true variance of noise effects.

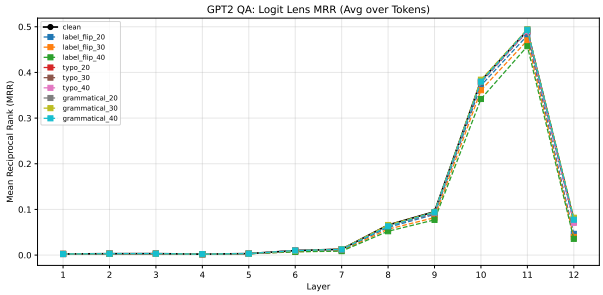
F.2 Clean-vs-Clean CKA Baseline

To establish a ceiling for CKA and confirm that noise-induced CKA drops are not attributable to seed variance alone, we compute CKA between clean models trained with different random seeds.

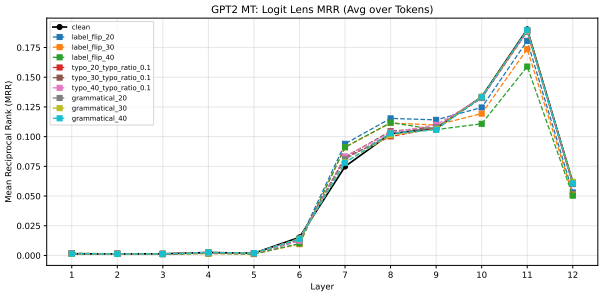
The clean-vs-clean CKA floor of 0.890 is far above the noise-condition values (0.11–0.42), confirming that the representational changes reported in our main experiments reflect genuine noise effects rather than stochastic training variance.

G Five-Token MRR Results

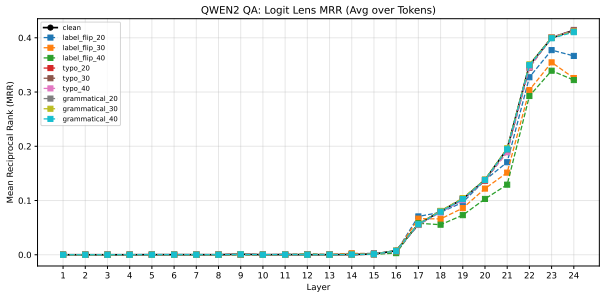
We extend the first-token MRR analysis from subsection 5.3 by computing the average MRR over the first five target tokens. At each layer, the hidden state is projected through the language model head, and we



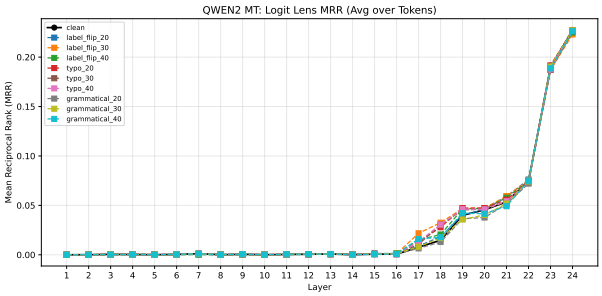
(a) QA - GPT-2 Small



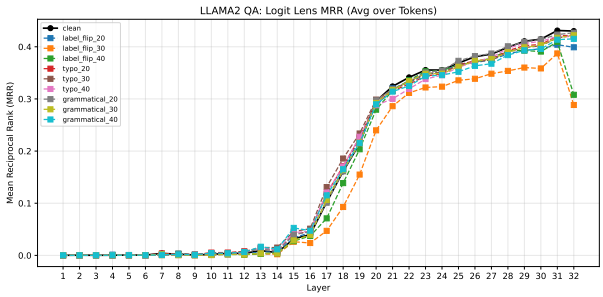
(b) MT - GPT-2 small



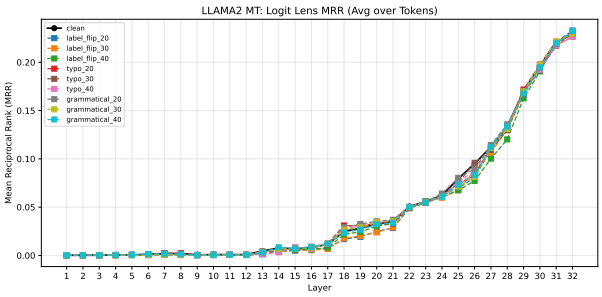
(c) QA - Qwen2-0.5B



(d) MT - Qwen2-0.5B



(e) QA - Llama2-7B



(f) MT - Llama2-7B

Figure 10: Layer-wise top-5 MRR for GPT-2 Small (124M), Qwen-2 (0.5B) and Llama-2 (7B).

compute the reciprocal rank for each of the five tokens independently using autoregressive generation. The results, shown in Figure 10, are consistent with the first-token MRR patterns reported in the main text.

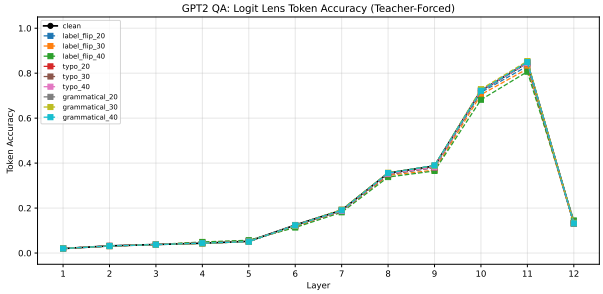
H Teacher Forced Five-Token MRR Results

For each evaluation sample, we compute token accuracy under a teacher-forced setting: at each of the first five target positions, the model receives the ground-truth prefix tokens and predicts the next token. Token accuracy at layer ℓ is defined as:

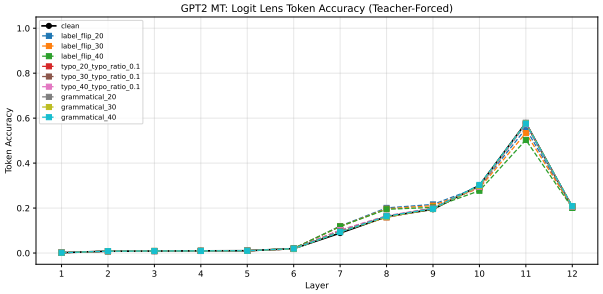
$$\mathrm{TokAcc}_\ell = \frac{1}{|S|} \sum_{s \in S} \frac{1}{5} \sum_{j=1}^{5} \mathbf{1}\left[p_\ell(\cdot \mid t^s_{<j}) = t^s_j\right] \tag{5}$$

where t^s_j is the j-th target token for the sample s and $p_\ell(\cdot \mid t^s_{<j})$ is the distribution obtained by projecting the layer-ℓ hidden state through the language model head.

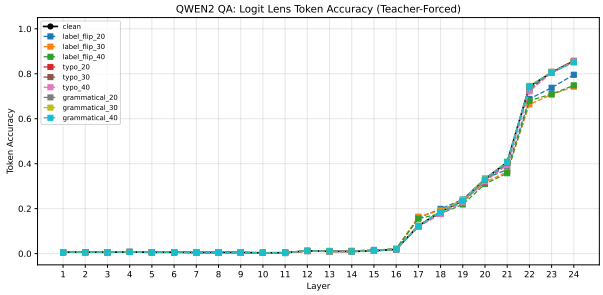
This metric complements MRR by providing a binary measure of prediction correctness rather than a rank-based measure, and is less sensitive to near-miss rankings. The results are shown in Figure 11.



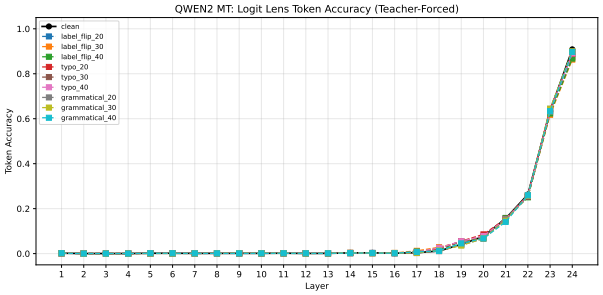
(a) QA - GPT-2 Small



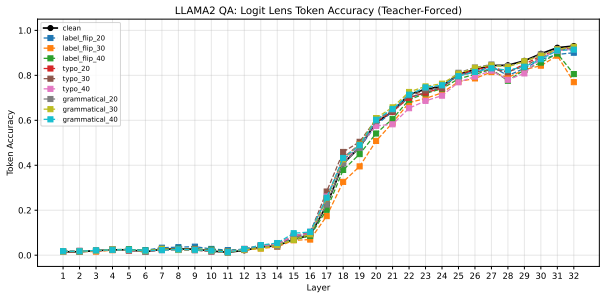
(b) MT - GPT-2 Small



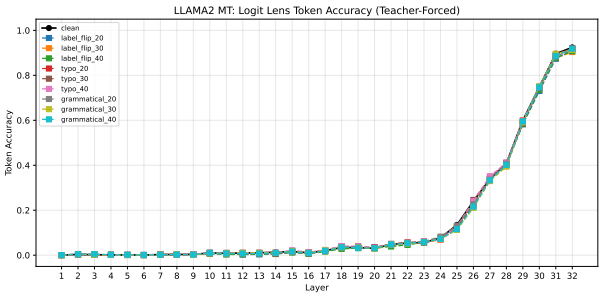
(c) QA - Qwen2-0.5B



(d) MT - Qwen2-0.5B



(e) QA - Llama2-7B



(f) MT - Llama2-7B

Figure 11: Layer-wise top 5 token accuracy for the 3 models on (a) question answering and (b) machine translation under all noise conditions. At each layer, hidden states are projected through the LM head, and accuracy is computed as the fraction of the first 5 generated tokens matching the reference.