A Comparative Study of Learning Paradigms in Large Language Models via Intrinsic Dimension

Saahith Janapati
University of Virginia
jax4zk@virginia.edu

Yangfeng Ji
University of Virginia
yj3fs@virginia.edu

Abstract

The performance of Large Language Models (LLMs) on natural language tasks can be improved through both supervised fine-tuning (SFT) and in-context learning (ICL), which operate via distinct mechanisms. SFT updates the model's weights by minimizing loss on training data, whereas ICL leverages task demonstrations embedded in the prompt, without changing the model's parameters. This study investigates the effects of these learning paradigms on the hidden representations of LLMs using Intrinsic Dimension (ID). We use ID to estimate the number of degrees of freedom between representations extracted from LLMs as they perform specific natural language tasks. We first explore how the ID of LLM representations evolves during SFT and how it varies due to the number of demonstrations in ICL. We then compare the IDs induced by SFT and ICL and find that ICL consistently induces a higher ID compared to SFT, suggesting that representations generated during ICL reside in higher dimensional manifolds in the embedding space. [1]

1 Introduction

Large Language Models (LLMs) have transformed the field of Natural Language Processing through their general natural language understanding capabilities, which can be applied to a broad range of tasks. The performance of an LLM on a specific task can be improved through two primary learning paradigms: supervised fine-tuning (SFT) and in-context learning (ICL). SFT adapts pretrained models to specific tasks by updating their parameters, while ICL requires no parameter updates, relying instead on task-specific demonstrations within the model's context window. Despite their widespread success, how these methods influence a model's internal representation space is still not fully understood.

Intrinsic dimension (ID) is a useful metric for assessing the geometric complexity of a model's representations. It quantifies the number of degrees of freedom in the representation space, serving as a measure of the complexity of the underlying manifolds where the embeddings reside.

In this work, we analyze the intrinsic dimension (ID) of hidden representations across model layers during task execution under both supervised fine-tuning (SFT) and in-context learning (ICL). Specifically, we explore:

- How fine-tuning duration influences ID of representations on both training and validation data.
- How the number of demonstrations used in ICL affects ID of representations.

Our findings reveal that (1) the ID sometimes decreases during the early stages of fine-tuning but generally increases in the later stages, and (2) the ID increases initially with more demonstrations in ICL, then either plateaus or decreases as the number of demonstrations continues to rise.

We then conduct experiments directly comparing the intrinsic dimensions of ICL and fine-tuning across several models and datasets. We find that the intrinsic dimensions of representations from fine-tuned models are generally lower than those from models using ICL, even though the fine-tuned models achieve higher accuracy than the ICL models. Additionally, our results suggest that ID may serve as a practical heuristic for selecting the optimal number of demonstrations in ICL to maximize performance while minimizing input length. These findings shed light on the differing impacts that the two learning paradigms have on the representation space of LLMs.

[1] Code is available at the following GitHub repo.

Proceedings of the 10th Workshop on Representation Learning for NLP (RepL4NLP-2025), pages 59–86
May 4, 2025 ©2025 Association for Computational Linguistics

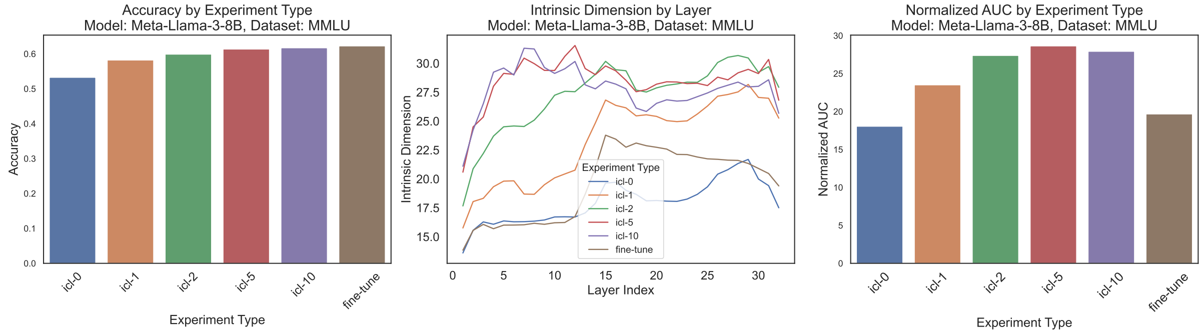


Figure 1: Accuracy, intrinsic dimension, and normalized AUC for the Llama-3-8B model on the MMLU dataset. (a) Fine-tuning achieves the highest accuracy. (b) ICL produces intermediate representations with higher intrinsic dimensions across model layers compared to zero-shot (ICL-0) and fine-tuned models. (c) Normalized AUC increases with the number of demonstrations in ICL, while fine-tuned models exhibit lower AUC.

2 Background

2.1 Decoder Transformer Architecture

LLMs are built on the Transformer decoder architecture, which processes token sequences through a series of Transformer layers. In each layer, token representations are updated via self-attention that considers only the preceding tokens from the previous layer, progressively encoding information for the next-token prediction task. The final layer then uses the representation of the last token to predict the next token in the sequence. In this work, we analyze the intrinsic dimension of the representations corresponding to the last token of sequences where LLMs are prompted to perform specific natural language tasks.

2.2 Intrinsic Dimension Estimation

Intrinsic dimension (ID) refers to the minimal number of variables required to capture the essential structure of high-dimensional data. Although modern neural networks operate in high-dimensional spaces (e.g., the hidden representations of Llama-3-8B span 4096 dimensions), the representations corresponding to a specific dataset or task often lie on a manifold of much lower dimension. This occurs because the network disentangles and extracts the most relevant lower-dimensional features needed to complete the task.

According to the manifold hypothesis, real-world data typically resides on a low-dimensional manifold (Goodfellow, 2016). Therefore, to effectively solve tasks—such as next-token prediction—neural networks must learn representations that align with this low-dimensional structure. Consequently, the intrinsic dimension of data representations provides unique insight into the complexity of the representation spaces constructed across the layers of a neural network.

In this work, we estimate the intrinsic dimension (ID) of our representations using the **TwoNN estimator**, as introduced by Facco et al. (2017). We chose this method because of its simplicity, computational efficiency, and robustness when handling datasets with non-uniform densities and high-dimensional curvature—common challenges in neural network representations.

The TwoNN estimator operates on a set of points by computing the distances to each point's first (r_1) and second (r_2) nearest neighbors. For a given point x, the ratio

$$\mu = \frac{r_2}{r_1}$$

is calculated. The intrinsic dimension d is then derived from the empirical cumulative distribution function (CDF) of μ. Specifically, the log-linear relationship between $\log(\mu)$ and $\log\big(1 - F_{\text{emp}}(\mu)\big)$, where $F_{\text{emp}}(\mu)$ is the empirical CDF, is used to estimate d:

$$d = -\frac{\log\big(1 - F_{\text{emp}}(\mu)\big)}{\log(\mu)}$$

The TwoNN estimator has been successfully applied in several prior works analyzing the intrinsic dimension of neural network representations, including Sharma and Kaplan (2022), Ansuini et al. (2019), Valeriani et al. (2024), and specifically in large language models (LLMs) by Cheng et al. (2023) and Lee et al. (2024). We also validate the correlation between the TwoNN estimator and another widely used intrinsic dimension estimator—the Maximum Likelihood Estimator introduced by

Levina and Bickel (2004)—in Appendix F as a sanity check.

3 Related Works

3.1 Supervised Fine-Tuning in LLMs

Pre-trained LLMs can be quickly adapted to improve performance on natural language tasks through supervised fine-tuning, which updates the model's parameters via gradient descent on task-specific training examples.

Aghajanyan et al. (2020) show that fine-tuning large language models often requires updating only a low-dimensional subspace of parameters to achieve near-optimal performance. (Note that their work focuses on the intrinsic dimension of the parameter space, whereas our work examines the intrinsic dimension of the representation space.) Building on this, Hu et al. (2021) introduce Low-Rank Adaptation (LoRA), a method that injects low-rank matrices into the weight matrices for fine-tuning instead of updating all parameters. We employ LoRA for all our fine-tuning experiments.

3.2 In-Context Learning

Introduced in GPT-3 by Brown (2020), ICL (or few-shot learning) refers to the ability of LLMs to learn to perform a task in a single forward pass, using (input, output) pairs embedded in a prompt.

Dai et al. (2022) provides evidence that ICL operates as implicit meta-optimization, where GPT models perform a gradient-like update via attention mechanisms during the forward pass. This suggests that ICL replicates fine-tuning behavior; specifically, they demonstrate that attention outputs and weights are updated in a direction similar to that of fine-tuning.

Xie et al. (2021) explain in-context learning as implicit Bayesian inference, where large language models infer latent document-level concepts during pretraining. These inferred concepts are then leveraged at test time to solve tasks based on the input-output examples provided in prompts.

Expanding the ICL paradigm to long-context models, Agarwal et al. (2024) studied many-shot ICL, in which hundreds or thousands of task examples are used to improve the performance of frontier models. Their work finds that an increasing number of demonstrations generally improves model performance on a variety of complex tasks, such as mathematical problem-solving.

3.3 Intrinsic Dimension in Deep Learning

Ansuini et al. (2019) investigated the intrinsic dimensionality (ID) of data representations across various convolutional neural networks (CNNs) for image classification. They observed a consistent "hunchback" pattern in ID evolution—an initial increase in the early layers followed by a progressive decrease in later layers.

Valeriani et al. (2024) extended this analysis to protein language models and image transformers, finding that the evolution of representations across layers of these models is also marked by distinct phases of ID growth and compression.

Yin et al. (2024) explore the use of Local Intrinsic Dimension (LID) to detect untruthful outputs from LLMs. Their study reveals that truthful outputs typically exhibit lower LIDs compared to hallucinated ones, suggesting that LID can serve as a signal for truthfulness in LLM generations. They also identify a positive relationship between the ID of data representations and validation performance during fine-tuning.

Cheng et al. (2023) demonstrate that intrinsic dimension correlates with fine-tuning ease and perplexity, with low-dimensional representations enabling faster task adaptation. Moreover, they find that ID values are consistent across model sizes, supporting the manifold hypothesis and suggesting that LLMs trained on similar data recover comparable intrinsic dimensions.

Of particular relevance to our study is the concurrent work of Doimo et al. (2024), which examines the internal representations of LLMs solving tasks from the MMLU dataset using both ICL and SFT. Their analysis reveals that ICL forms semantic clusters in the early layers, while SFT sharpens these clusters in later layers for task-specific answers. Moreover, they observe that intrinsic dimension (ID) increases with a higher number of demonstrations in ICL, and that SFT generally induces a higher ID compared to ICL. In contrast, our findings indicate that beyond a certain range of ICL demonstrations, ID may plateau or even decrease, and that SFT consistently induces a lower ID than ICL.

To our knowledge, our work is the first to systematically analyze and compare intrinsic dimension across the two learning paradigms for numerous datasets and models. We further provide in-depth analyses of how ID is affected by various factors within each paradigm, such as the number of gradient steps in SFT and the number of demonstrations in ICL.

3.4 Intrinsic Dimension and Neural Network Scaling Laws

Sharma and Kaplan (2022) propose that the power-law scaling of neural network performance is rooted in the intrinsic dimensionality (ID) of the data manifold. They empirically demonstrate that the ID of learned representations, particularly in the final hidden layer, directly relates to the scaling exponent. Their theory, predicting a scaling exponent of approximately $\alpha \approx 4/d$ (where d is ID), suggests that neural networks achieve efficient scaling by effectively performing regression on a lower-dimensional data manifold, thus linking model capacity to the data's inherent complexity.

4 Methods

We perform experiments using subsets from the following datasets: AG News (Zhang et al., 2015), CoLA (Warstadt et al., 2018), CommonsenseQA (Talmor et al., 2018), MMLU (Hendrycks et al., 2020), MultiNLI (Williams et al., 2017), QNLI (Wang, 2018), QQP (Wang et al., 2017), and SST2 (Socher et al., 2013).

For these experiments, we utilize the following open-source LLMs: Llama-3-8B (Dubey et al., 2024), Llama-2-13b, Llama-2-7b (Touvron et al., 2023), and Mistral-7B-v0.3 (Jiang et al., 2023), running them on 6 NVIDIA A6000s.

For each dataset, we created a training set of 1000 examples and a validation set of 5000 examples. We use the 5000 validation examples to ensure stability of the TwoNN estimator. Details regarding dataset creation can be found in Appendix G. Details of split generations and prompt templates are provided in Appendix G.

We calculate the accuracy of model responses using the logit probabilities assigned to the tokens corresponding to the possible answers for each question. We mark a response as correct if the probability corresponding to the first token of the correct answer label is the highest.

4.1 Computing Intrinsic Dimension

In both Supervised Fine-Tuning (SFT) and In-Context Learning (ICL) paradigms, a language model receives an input sequence of tokens and is tasked with generating an output sequence that answers the given prompt. To quantify the intrinsic dimensionality (ID) of a model's representations for a given dataset, we extract the hidden state activations at each layer of the LLM. Specifically, we focus on the activations corresponding to the **last token of each input sequence** in the dataset. For a model with L layers and a dataset containing N input sequences, this process yields L sets of hidden state representations. Each set corresponds to a specific layer and comprises N representation vectors (one for each input sequence in the dataset). Subsequently, we compute the intrinsic dimension (ID) for each of these L sets of N vectors. This provides us with an ID estimate for the representation space at each layer. By plotting the Layer Index against the corresponding ID estimates, we construct what we term the **Intrinsic Dimension Curve**.

To derive a single, aggregated metric that encapsulates the intrinsic dimensionality across all layers of a model, we calculate the **Normalized Area Under the Curve (AUC)** of the Intrinsic Dimension Curve, defined as follows:

$$\text{Normalized AUC} = \frac{1}{L} \sum_{i=1}^{L-1} \frac{1}{2} \left(\text{ID}_i + \text{ID}_{i+1}\right)$$

In this equation, ID_i denotes the intrinsic dimension estimate at layer i. The formula employs the trapezoidal rule for numerical integration to approximate the area beneath the Intrinsic Dimension Curve. The normalization by L (the number of layers) enables fair comparisons of intrinsic dimensionality across models with varying depths.

5 Dynamics of ID during Supervised Fine-Tuning

5.1 Supervised Fine-Tuning Experimental Setup

To investigate the impact of supervised fine-tuning at a granular level, we conduct experiments using the 8 datasets discussed in Section 4 and the Llama-3-8B and Llama-2-13B models.

Using the training split for each of the datasets, we perform LoRA fine-tuning on the query, key, value, and output projection matrices of attention heads across all layers of the model. For all models, we fine-tune with a batch size of 16 for 15 epochs. For all fine-tuning runs, we use LoRA hyperparameters of $r = 64$, lora_alpha $= 16$, lora_dropout $= 0.1$, no LoRA bias, and a learning rate of $1e^{-4}$.

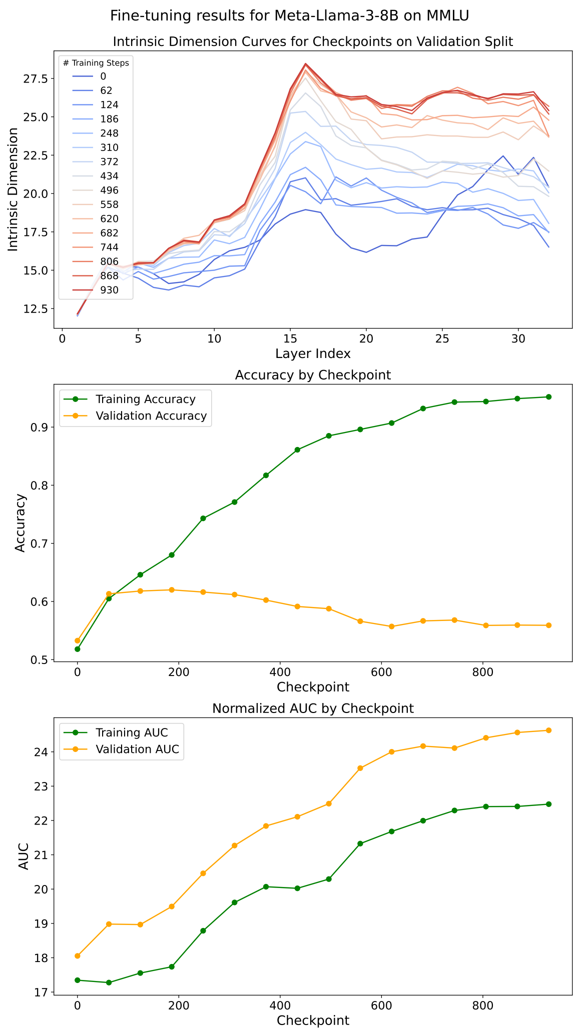


Figure 2: Fine-tuning results for Meta-Llama-3-8B on the MMLU dataset. (a) Intrinsic Dimension curves on the validation split increase across training steps. (b) Training accuracy improves steadily, while validation accuracy plateaus. (c) Normalized AUC for training and validation sets increases throughout fine-tuning.

During the fine-tuning process for a specific model and dataset, we save a checkpoint of the model after every epoch (~62 gradient update steps). For each checkpoint, we evaluate the model's accuracy and measure the intrinsic dimension (ID) of the hidden representations on prompts from the training and validation splits for the dataset.

5.2 Intrinsic Dimension Generally Increases Through Fine-Tuning

As depicted in Figure 2c, we find that ID of representations corresponding to both training data and validation data sometimes decreases during the initial stages of fine-tuning, but then generally increases as fine-tuning progresses.

We also observe larger changes in ID values for later layers of the models, despite LoRA adaptation being applied on all the layers with the same configuration (Figure 2a).

Additionally, we find that the AUC values of the model on the training set and validation set are often highly correlated with each other during the training process (Figure 2c). Experimental results for all models and datasets can be found in Appendix B.

Prior work by Yin et al. (2024) found that on Question-Answering datasets, intrinsic dimension of representations is correlated with validation performance and can therefore be used as a heuristic to select final checkpoints. In general, we do not find this trend to hold on the datasets and models we tested. In fact, as shown in Figure 13, large increases in validation accuracy sometimes coincide with drops in ID on both the training and validation datasets.

6 Relationship of ID in ICL with Different Numbers of Shots

6.1 In-Context Learning Experimental Setup

To investigate the impact of ICL on the ID of model representations, we conduct experiments using the Llama-3-8B and Llama-2-13B models. The datasets included in our evaluation are CommonsenseQA, MMLU, and QNLI.

We evaluate ICL performance using various values of k, where k denotes the number of demonstrations in the ICL prompt. The values considered are $k \in \{0, 1, 2, 5, 10, 12, 14, 16, 18, 20\}$. Note that $k = 0$ serves as a baseline, representing the model's performance in the absence of both ICL and SFT.

For each k and dataset, we generate 5000 ICL prompts (one for each element of the validation split of the dataset). Each ICL prompt includes k unique demonstrations, or (input, output) pairs, randomly sampled from the training set. While we ensure that demonstrations within a single prompt are unique, they may be reused across different prompts.

6.2 ID Has a Non-Linear Relationship with Number of Demonstrations

We observe that ID values across layers can fluctuate until a threshold value of k (typically around 5 to 10 for most model configurations), after which

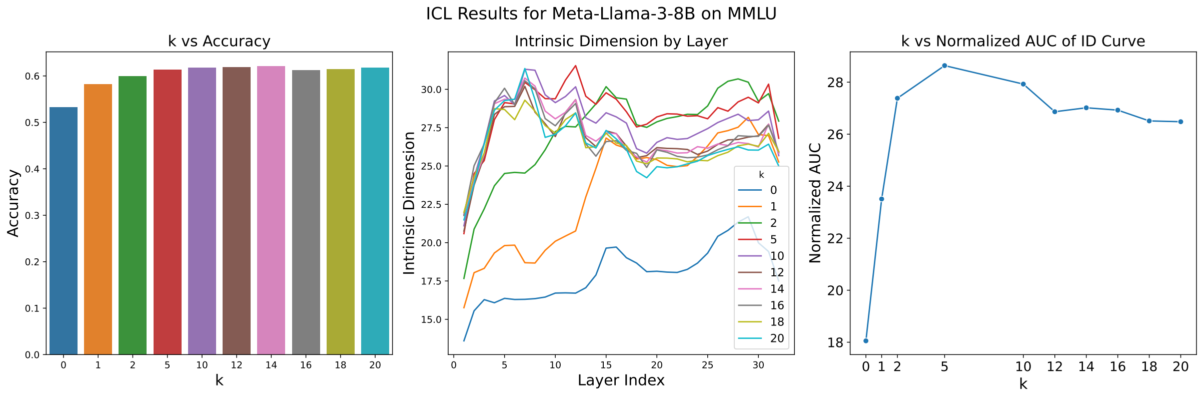


Figure 3: (ICL) results for Meta-Llama-3-8B model on MMLU dataset. (a) Accuracy increases, then plateaus as number of demonstrations increases (b) Intrinsic Dimension (ID) curves for different values of k. (c) Normalized AUC of the ID curves peaks at k=5, which also aligns with saturation of accuracy.

they either plateau or steadily decrease for larger values of k (see Figure 3c). Results for all model and dataset configurations are provided in Appendix A. This observation extends the findings of Doimo et al. (2024), who found that ID increased as k was varied from 0, 1, 2, and 5, by demonstrating that beyond a certain number of demonstrations, the trend can reverse.

We observe that across most (model, dataset) combinations, the shapes of the intrinsic dimension (ID) curves correlate strongly with each other for $k \geq 2$.

Due to our procedure of selecting demonstrations with replacement, we suspected that the plateau in ID for larger values of k might be caused by a greater number of demonstrations shared across prompts. We hypothesized that shared demonstrations could make representations corresponding to these prompts artificially similar, thereby skewing ID results. To test this, we performed additional experiments using a larger number of dataset elements from the CommonsenseQA, QNLI, and AG News datasets, which contain enough training elements to ensure that demonstrations are not reused in prompts for more than one element of the validation set. We observed the same trend—an increase followed by a general plateau in the ID—suggesting that the plateau is likely not due to the reuse of demonstrations among the prompts. Full results for this experiment can be found in Appendix D.

Furthermore, we find that peaks in the k versus AUC relationship align with peak (or near-peak) accuracy in 5 out of the 6 ICL experiments we conducted. Thus, the k value corresponding to the peak ID may serve as a practical indicator of the optimal number of demonstrations to use for ICL, maximizing performance while minimizing input length.

One hypothesis for why ID plateaus or slightly decreases as k increases is that more demonstrations allow the model to more effectively capture the underlying task conveyed by the demonstrations, causing representations corresponding to different inputs to become more similar. This idea is supported by previous theoretical analysis of ICL by Xie et al. (2021), which posits that a greater number of demonstrations helps the model more effectively infer the latent concept across demonstrations.

Finally, we find that across most experiments, accuracy either steadily increases or plateaus with higher numbers of demonstrations (Figure 3a).

7 Comparing Intrinsic Dimension of In-Context Learning and Supervised Fine-Tuning

7.1 Experiment Setup for Comparative Analysis

We conduct a series of experiments to directly compare the ID curves obtained from both SFT and ICL, following similar setups as discussed in Sections 5 and 6. For the fine-tuning experiments in this section, we train for only 4 epochs and measure the accuracy and ID solely at the final checkpoint. This choice is motivated by the observation in Section 5 that models tend to overfit beyond 4 epochs across the tested datasets.

For the ICL experiments, we consider values of $k \in \{0, 1, 2, 5, 10\}$ for the number of demonstra-

Dataset	ICL-0	ICL-1	ICL-2	ICL-5	ICL-10	Finetune 1K
SST-2	0.685	0.633	0.731	0.807	0.832	**0.944**
CoLA	0.720	0.723	0.735	0.746	0.742	**0.750**
QNLI	0.517	0.513	0.555	0.590	0.585	**0.761**
QQP	0.417	0.462	0.485	0.508	0.519	**0.707**
MNLI	0.374	0.367	0.387	0.414	0.431	**0.676**
AGNews	0.638	0.573	0.712	0.772	0.809	**0.881**
CommonsenseQA	0.199	0.375	0.417	0.470	0.492	**0.500**
MMLU	0.449	0.488	0.511	0.524	0.531	**0.542**

Table 1: Average accuracy results for Datasets across ICL and SFT settings. SFT obtains the highest average accuracy for all datasets. Accuracy increases and then plateaus for higher number of demonstrations.

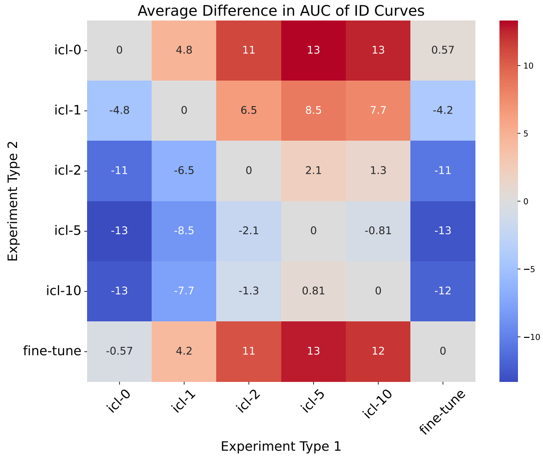


Figure 4: Heatmap showing the average differences in normalized AUC of ID curves between pairs of learning paradigms. Each value represents the average difference (Experiment Type 1 - Experiment Type 2), computed across all (model, dataset) pairs.

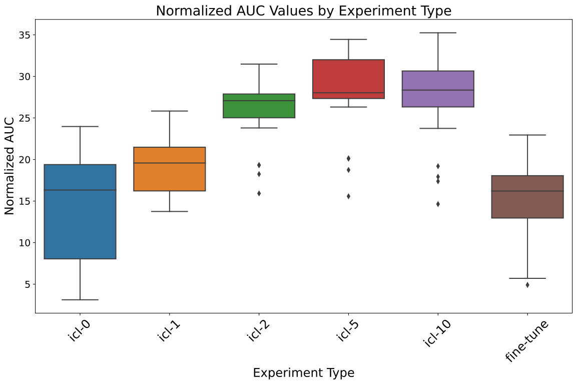


Figure 5: Boxplot displaying the distribution of normalized AUC values for different learning paradigms. Each point corresponds to the normalized AUC value for a (model, dataset) pair. The median normalized AUC peaks with 5-shot ICL, while values for SFT are closer to the 0-shot baseline (icl-0).

tions. These values are popular in practice, and our previous experiments in Section 6 indicate that ID curves tend to plateau when $k \geq 10$. We perform these experiments on all 8 datasets and 4 models discussed in Section 4.

7.2 In-context Learning Induces Higher IDs Compared to Fine-Tuning

We find that across all datasets and models, ICL prompts with $k \geq 5$ consistently induces higher intrinsic dimensions (IDs) across all layers compared to both SFT and 0-shot prompts (see Figures 1b and 1c). This contrasts with the findings of Doimo et al. (2024), who find that SFT models often induces higher ID than models performing ICL.

We also find that the ID values of models fine-tuned with 1000 samples tend to remain similar to the original ID of the baseline model on a zero-shot prompt (designated by icl-0). We present a heatmap displaying the average differences in normalized AUC between learning paradigms in Figure 4, and a boxplot depicting the distribution of normalized AUC values for the different paradigms in Figure 5.

7.3 Analysis of Intrinsic Dimension Curves

7.3.1 Differing Shapes of Intrinsic Dimension Curves

We observe that the exact shape of the Intrinsic Dimension curves is highly dependent on the dataset. For some datasets, such as AG News, we observe a consistent "hunchback" shape, where the ID initially increases and then is progressively lower in the later layers of the model across all models and learning paradigms (Figure 36). This shape has been reported by previous work (Yin et al., 2024) in QA datasets. However, this pattern does not consistently hold across all models, datasets, and learning paradigms. For example, on the QQP dataset, we

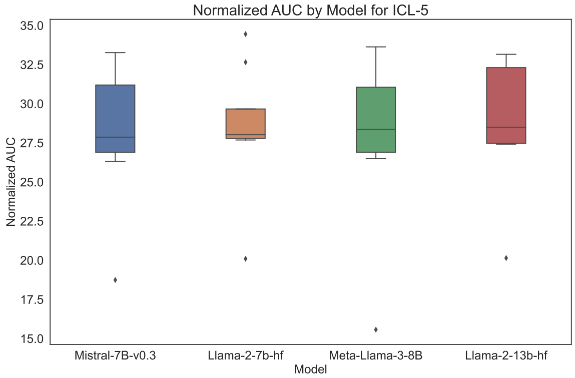


Figure 6: Boxplot displaying the distribution of normalized AUC values across datasets for each model in the ICL-5 shot setting. Each point corresponds to a (model, dataset) pair. The ID values lie in a narrow range, highlighting similarity in representation spaces across models.

do not observe a consistent hunchback shape for icl-0, icl-1, or fine-tuning learning paradigms (Figure 33). In contrast, prior work has shown that Convolutional Neural Networks (Ansuini et al., 2019), as well as Image Generation Transformers such as ImageGPT and Protein Language Models (Valeriani et al., 2024), exhibit consistent Intrinsic Dimension patterns across their layers for inputs of their respective data modalities. This difference suggests that LLMs encode data into more diverse manifolds in their representation space, potentially reflecting their generality and the complexity of their learning tasks compared to other neural networks.

We also find that, within a specific learning paradigm, the range of normalized AUC values across datasets is similar for the four different models we tested, despite the fact that these models come from different families and have different embedding dimensions (e.g., Llama-2-13b has a hidden dimension of 5120, while the other three models have hidden dimensions of 4096). Figure 6 depicts the range of normalized AUC values for the ICL-5 learning paradigm and shows that all values fall within a range of 20. We view this as evidence that different models may be generating representations with similar geometric complexity for a specific dataset, despite differences in model size or pre-training schemes. Similar boxplots for normalized AUC values from other experiments are included in Appendix B.2. These findings are in agreement with results from Cheng et al. (2023), which show that LLMs of different sizes and families create representations with similar ID values for a variety of text corpora.

7.4 Comparing Performance of Different Learning Paradigms

We found that models fine-tuned with 1k samples obtained the highest accuracy, while models performing ICL with 10 samples followed closely. This observation suggests that intrinsic dimension (ID) may not be directly related to accuracy: although fine-tuning with 1k samples yields ID values that remain closer to the baseline model, ICL models exhibit higher IDs yet achieve substantially lower accuracies. See Table 1 for the average performance of each learning paradigm across the models and datasets tested.

8 Summary

We present a detailed analysis of the intrinsic dimension (ID) induced by the SFT and ICL learning paradigms. Our experiments reveal that the normalized AUC of ID curves sometimes decreases during the initial stages of SFT but generally increases during the later stages.

Additionally, we observe that the normalized AUC of ID curves in ICL initially increases for small values of k (the number of demonstrations) but plateaus or slightly decreases as k increases further. Notably, the k value corresponding to the highest normalized AUC also achieves peak (or near-peak) accuracy, suggesting that ID may serve as a useful indicator for selecting the optimal number of demonstrations during ICL.

Finally, our direct comparison of ID curves from ICL and SFT reveals that representations generated during ICL consistently yield higher ID curves compared to those from SFT on 1k samples, even though SFT with 1k samples achieves the highest overall performance. This analysis provides evidence that the two learning paradigms induce distinct representational structures in the embedding space, with ICL representations occupying higher-dimensional manifolds.

9 Limitations

In this study, we limit our analysis to models with sizes between 7B and 13B parameters. Future work may extend this investigation to models of different sizes. We also focus on datasets defined by narrowly focused tasks and do not consider datasets with long-form answers. Due to computational constraints, we perform fine-tuning only using LoRA adapters and do not explore the impacts of full fine-tuning on intrinsic dimension.

References

Rishabh Agarwal, Avi Singh, Lei M Zhang, Bernd Bohnet, Stephanie Chan, Ankesh Anand, Zaheer Abbas, Azade Nova, John D Co-Reyes, Eric Chu, et al. 2024. Many-shot in-context learning. *arXiv preprint arXiv:2404.11018*.

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. 2020. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*.

Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. 2019. Intrinsic dimension of data representations in deep neural networks. *Advances in Neural Information Processing Systems*, 32.

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. 2022. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, Dublin, Ireland. Association for Computational Linguistics.

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

Emily Cheng, Corentin Kervadec, and Marco Baroni. 2023. Bridging information-theoretic and geometric compression in language models. *arXiv preprint arXiv:2310.13620*.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2022. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. *arXiv preprint arXiv:2212.10559*.

Diego Doimo, Alessandro Serra, Alessio Ansuini, and Alberto Cazzaniga. 2024. The representation landscape of few-shot learning and fine-tuning in large language models. *arXiv preprint arXiv:2409.03662*.

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*.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. 2017. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific reports*, 7(1):12140.

Ian Goodfellow. 2016. Deep learning.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jin Hwa Lee, Thomas Jiralerspong, Lei Yu, Yoshua Bengio, and Emily Cheng. 2024. Geometric signatures of compositionality across a language model's lifetime. *arXiv preprint arXiv:2410.01444*.

Elizaveta Levina and Peter Bickel. 2004. Maximum likelihood estimation of intrinsic dimension. *Advances in neural information processing systems*, 17.

Utkarsh Sharma and Jared Kaplan. 2022. Scaling laws from the data manifold dimension. *Journal of Machine Learning Research*, 23(9):1–34.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lucrezia Valeriani, Diego Doimo, Francesca Cuturello, Alessandro Laio, Alessio Ansuini, and Alberto Cazzaniga. 2024. The geometry of hidden representations of large transformer models. *Advances in Neural Information Processing Systems*, 36.

Alex Wang. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Fan Yin, Jayanth Srinivasa, and Kai-Wei Chang. 2024. Characterizing truthfulness in large language model generations with local intrinsic dimension. *arXiv preprint arXiv:2402.18048*.

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

A In-Context Learning Experiments

A.1 Llama-3-8B In-Context Learning Experiments

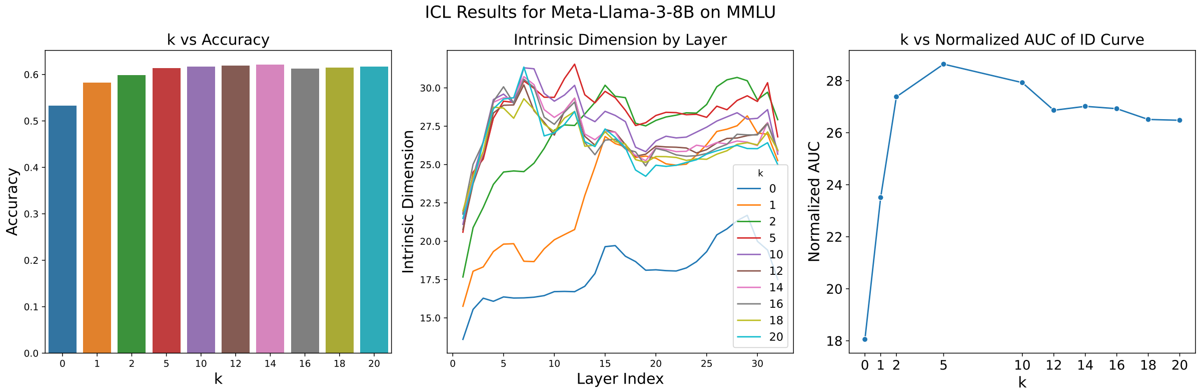


Figure 7: ICL Experiment Results for Meta-Llama-3-8B on MMLU

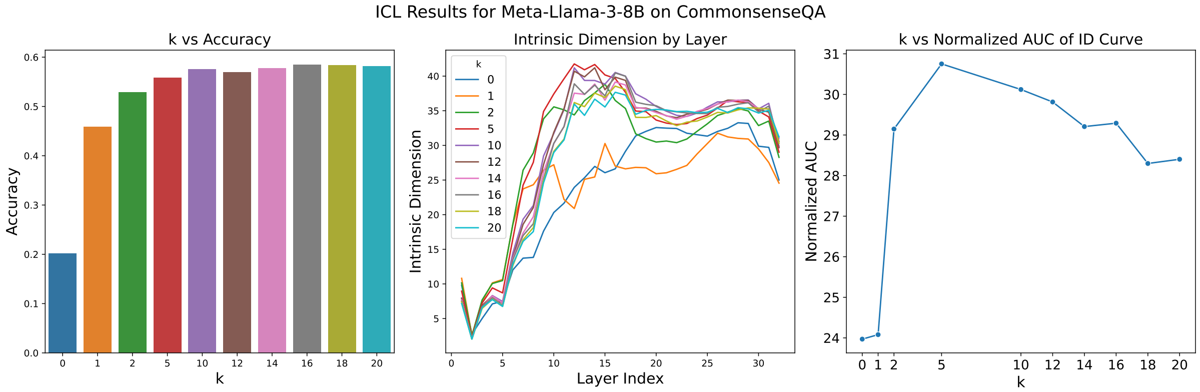


Figure 8: ICL Experiment Results for Meta-Llama-3-8B on CommonsenseQA

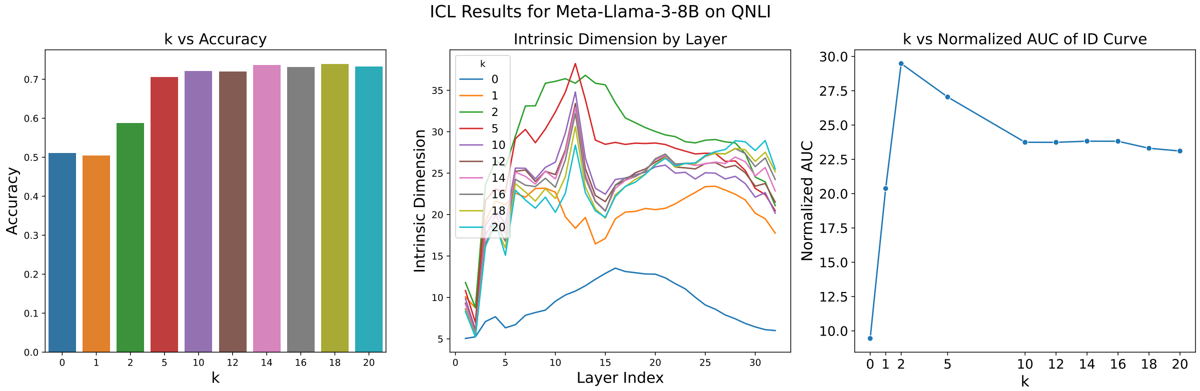


Figure 9: ICL Experiment Results for Meta-Llama-3-8B on QNLI

A.2 Llama-2-13b In-Context Learning Experiments

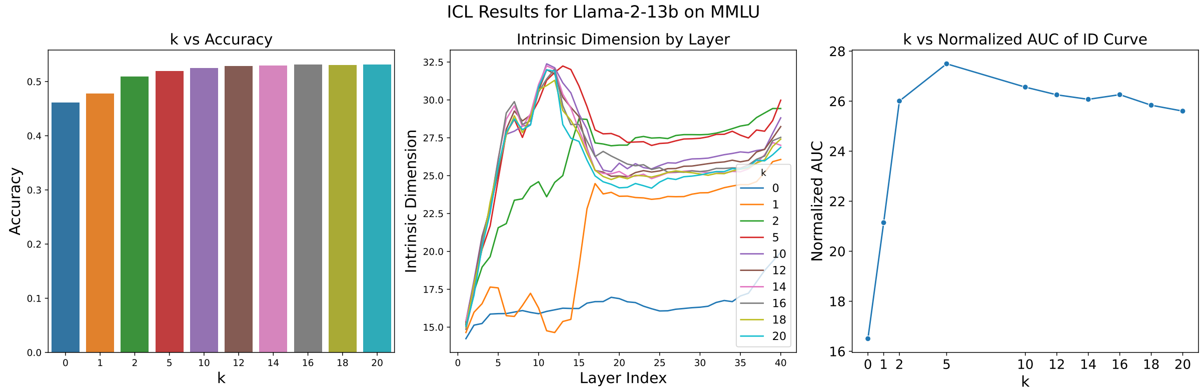


Figure 10: ICL Experiment Results for Llama-2-13b on MMLU

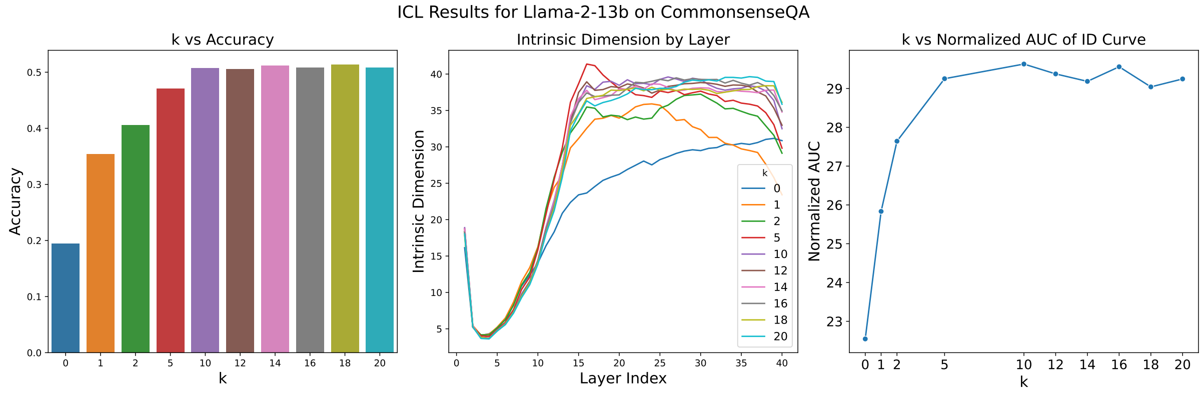


Figure 11: ICL Experiment Results for Llama-2-13b on CommonsenseQA

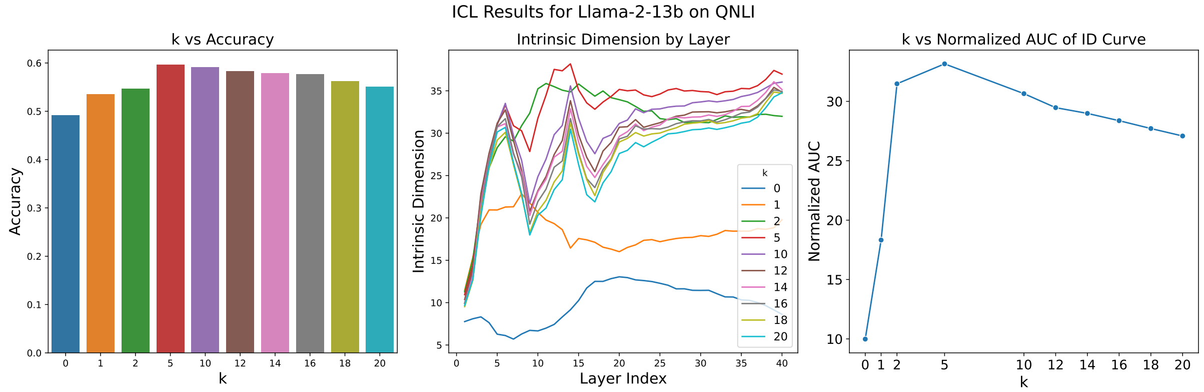


Figure 12: ICL Experiment Results for Llama-2-13b on QNLI

B Supervised Fine-Tuning Experiments

B.1 Supervised Fine-Tuning Results for Llama-3-8B

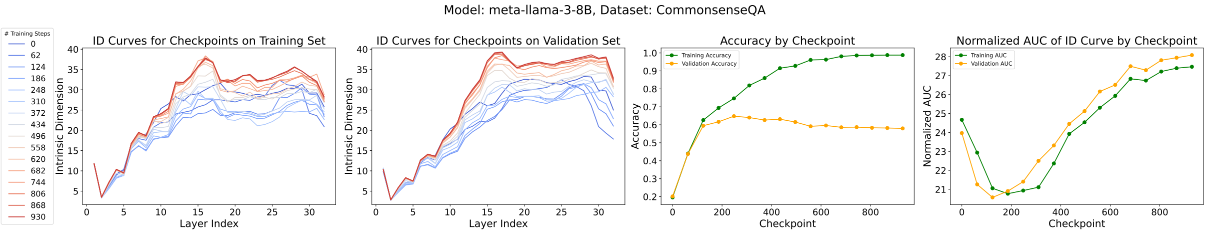


Figure 13: Supervised Fine-Tuning Results for Llama-3-8B on Commonsense QA

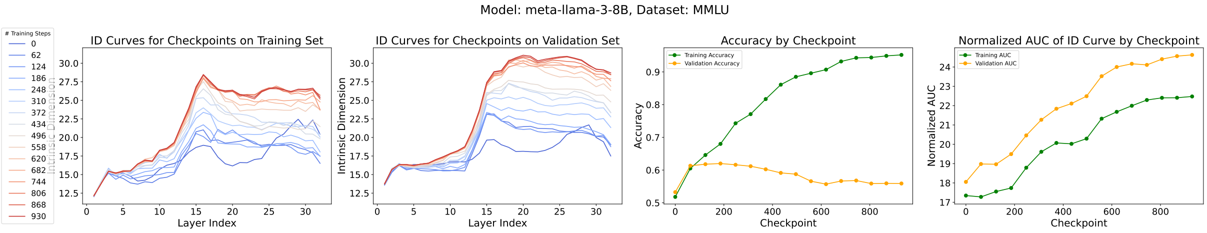


Figure 14: Supervised Fine-Tuning Results for Llama-3-8B on MMLU

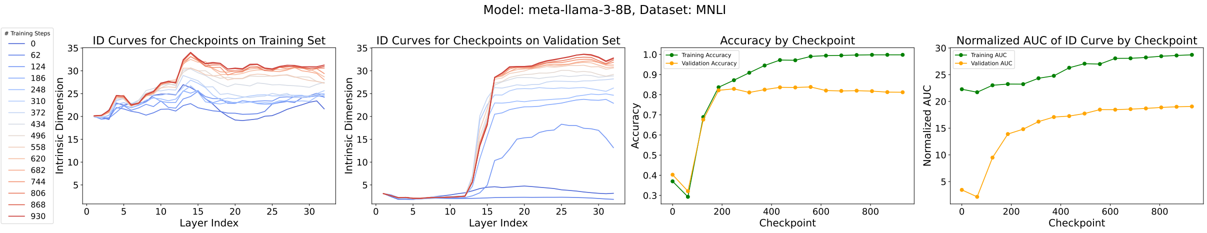


Figure 15: Supervised Fine-Tuning Results for Llama-3-8B on MNLI

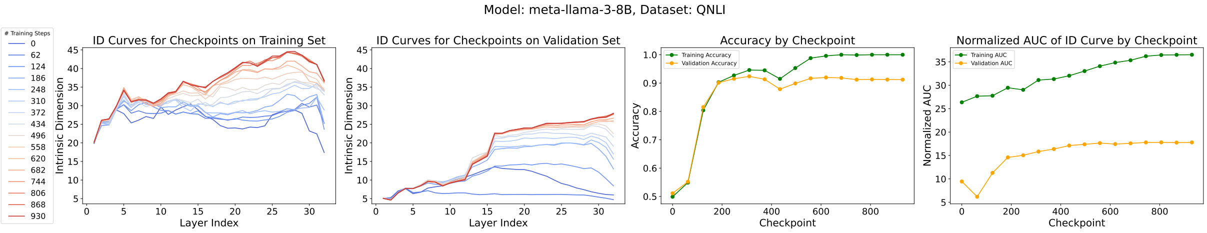


Figure 16: Supervised Fine-Tuning Results for Llama-3-8B on QNLI

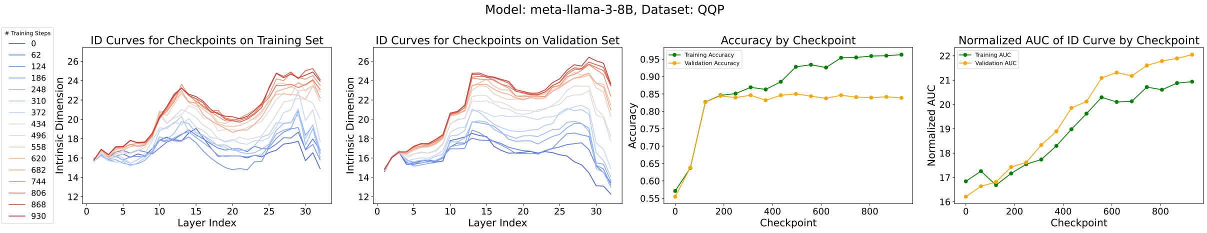


Figure 17: Supervised Fine-Tuning Results for Llama-3-8B on QQP

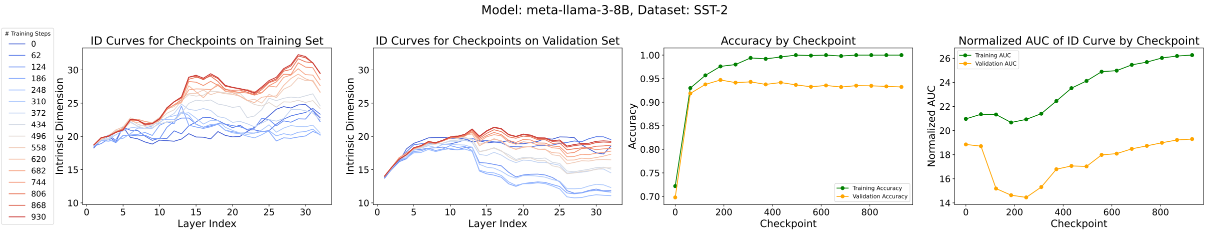


Figure 18: Supervised Fine-Tuning Results for Llama-3-8B on SST-2

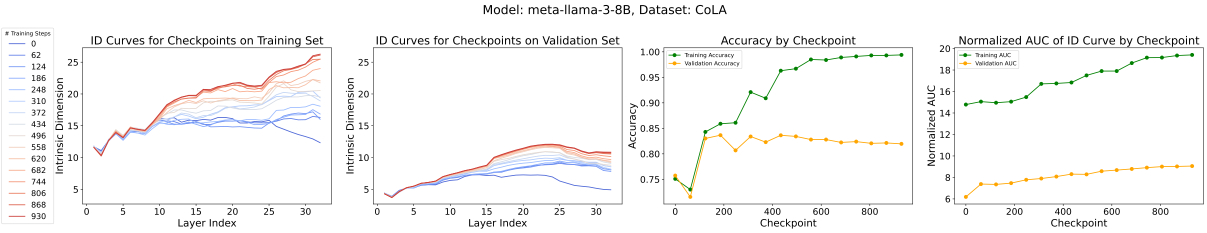


Figure 19: Supervised Fine-Tuning Results for Llama-3-8B on CoLA

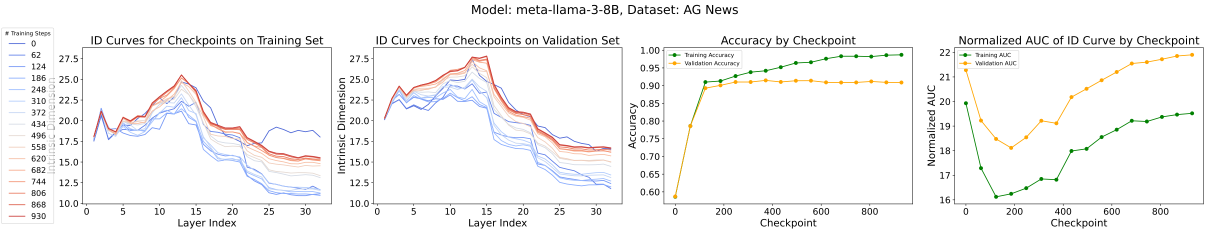


Figure 20: Supervised Fine-Tuning Results for Llama-3-8B on AG News

B.2 Supervised Fine-Tuning Results for Llama-2-13B

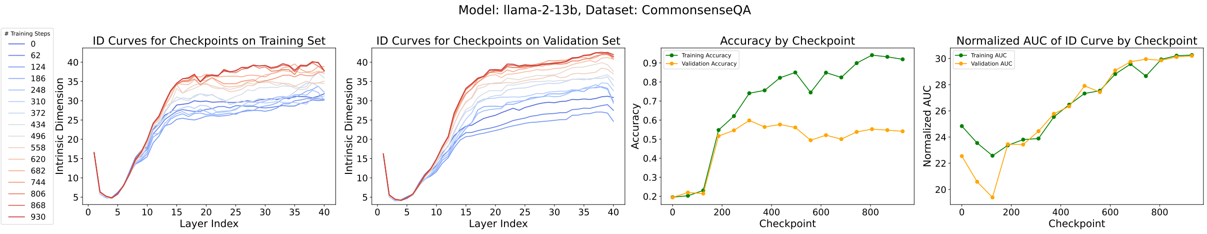


Figure 21: Supervised Fine-Tuning Results for Llama-2-13B on Commonsense QA

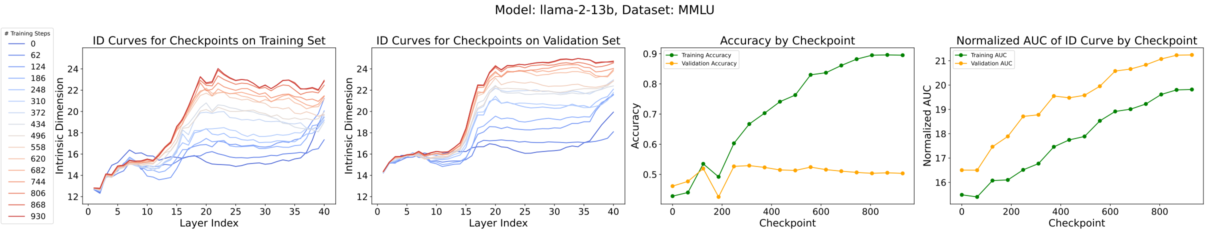


Figure 22: Supervised Fine-Tuning Results for Llama-2-13B on MMLU

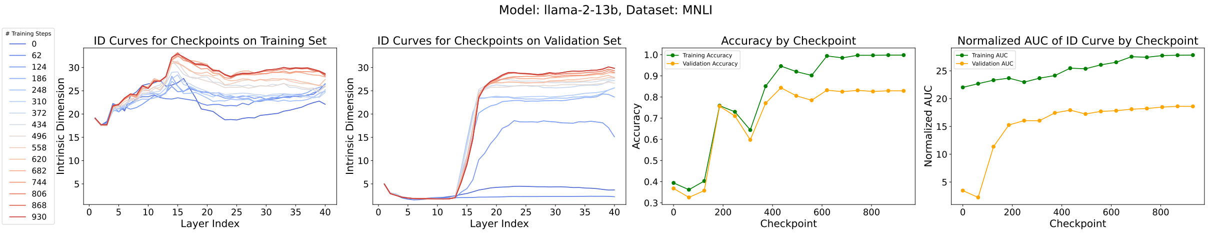


Figure 23: Supervised Fine-Tuning Results for Llama-2-13B on MNLI

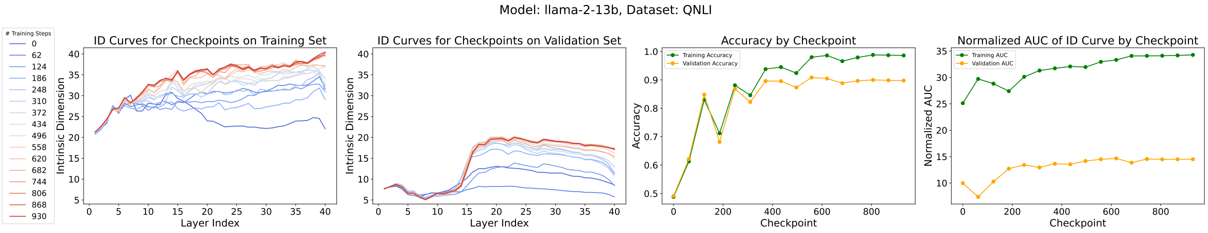


Figure 24: Supervised Fine-Tuning Results for Llama-2-13B on QNLI

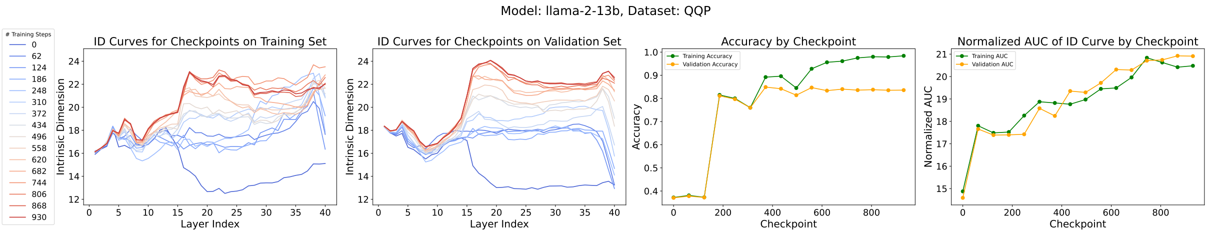


Figure 25: Supervised Fine-Tuning Results for Llama-2-13B on QQP

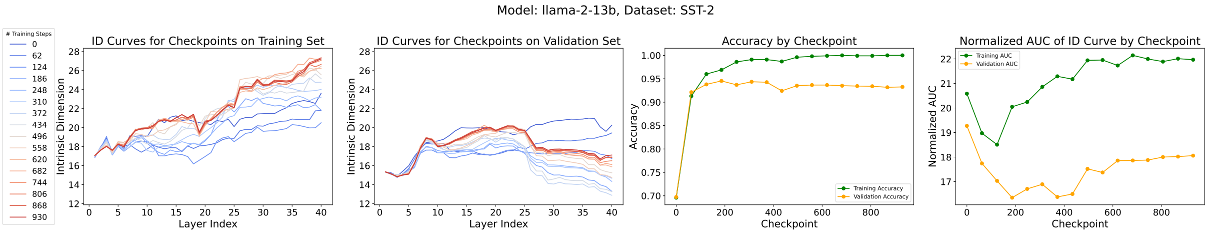


Figure 26: Supervised Fine-Tuning Results for Llama-2-13B on SST-2

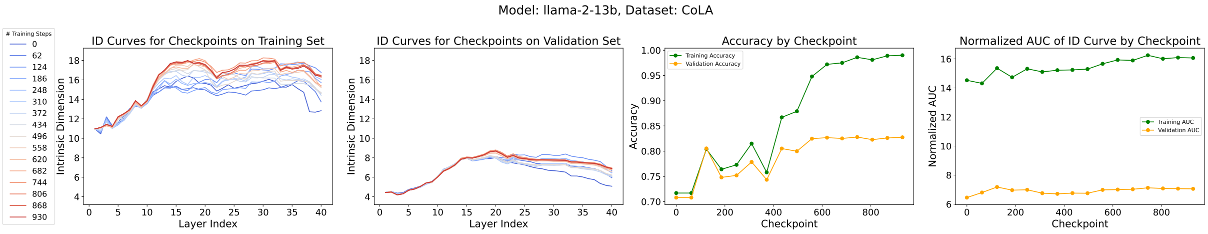


Figure 27: Supervised Fine-Tuning Results for Llama-2-13B on CoLA

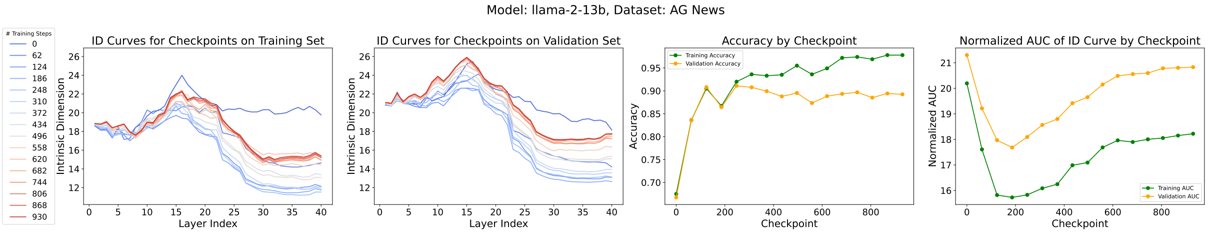


Figure 28: Supervised Fine-Tuning Results for Llama-2-13B on AG News

C Comparisons of Supervised Fine-Tuning and In-Context Learning

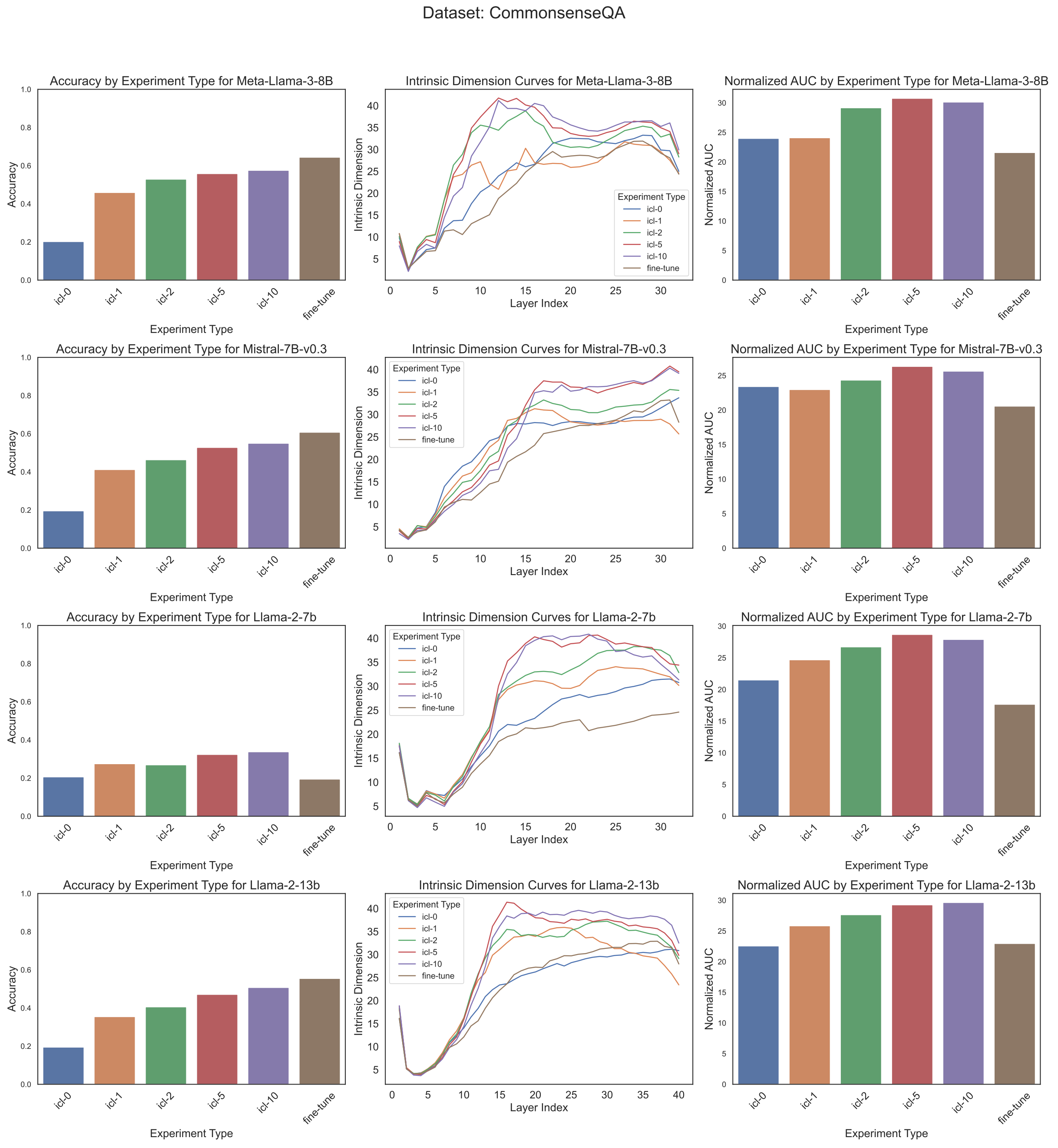
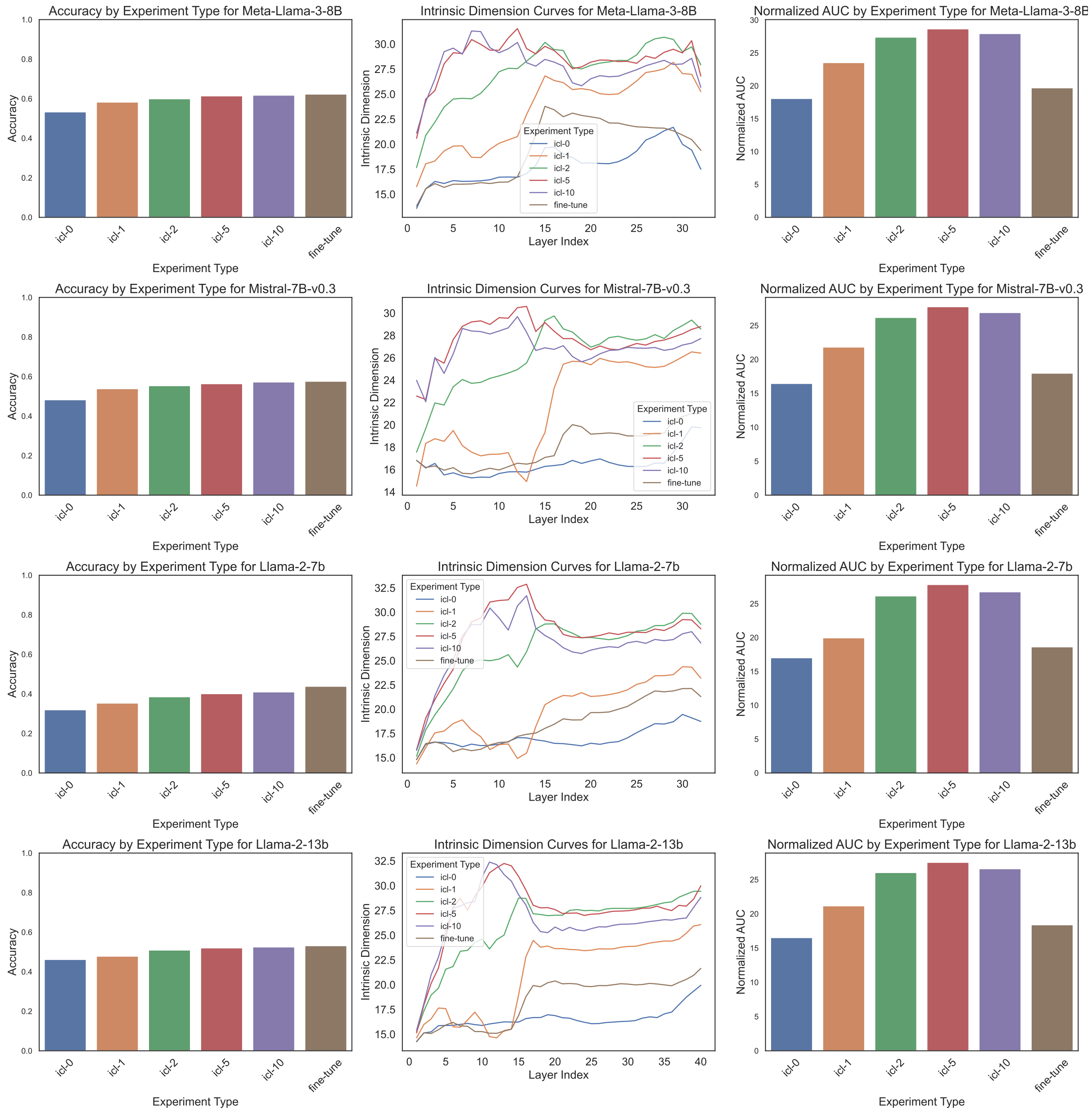


Figure 29: Comparison of Experimental Results for Commonsense QA

Dataset: MMLU


Accuracy by Experiment Type for Meta-Llama-3-8B

Intrinsic Dimension Curves for Meta-Llama-3-8B

Normalized AUC by Experiment Type for Meta-Llama-3-8B

Accuracy by Experiment Type for Mistral-7B-v0.3

Intrinsic Dimension Curves for Mistral-7B-v0.3

Normalized AUC by Experiment Type for Mistral-7B-v0.3

Accuracy by Experiment Type for Llama-2-7b

Intrinsic Dimension Curves for Llama-2-7b

Normalized AUC by Experiment Type for Llama-2-7b

Accuracy by Experiment Type for Llama-2-13b

Intrinsic Dimension Curves for Llama-2-13b

Normalized AUC by Experiment Type for Llama-2-13b

Figure 30: Comparison of Experimental Results for MMLU

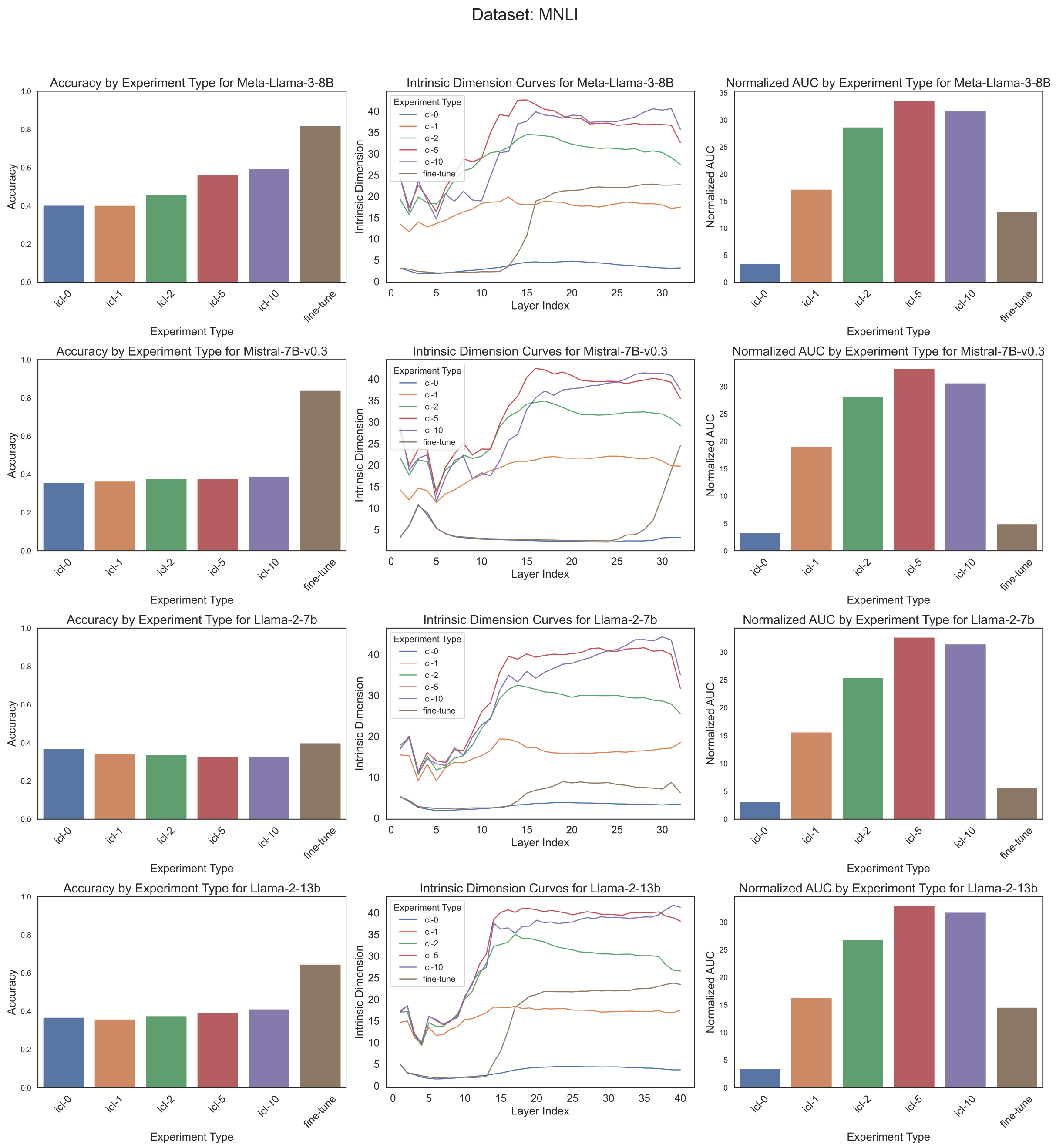
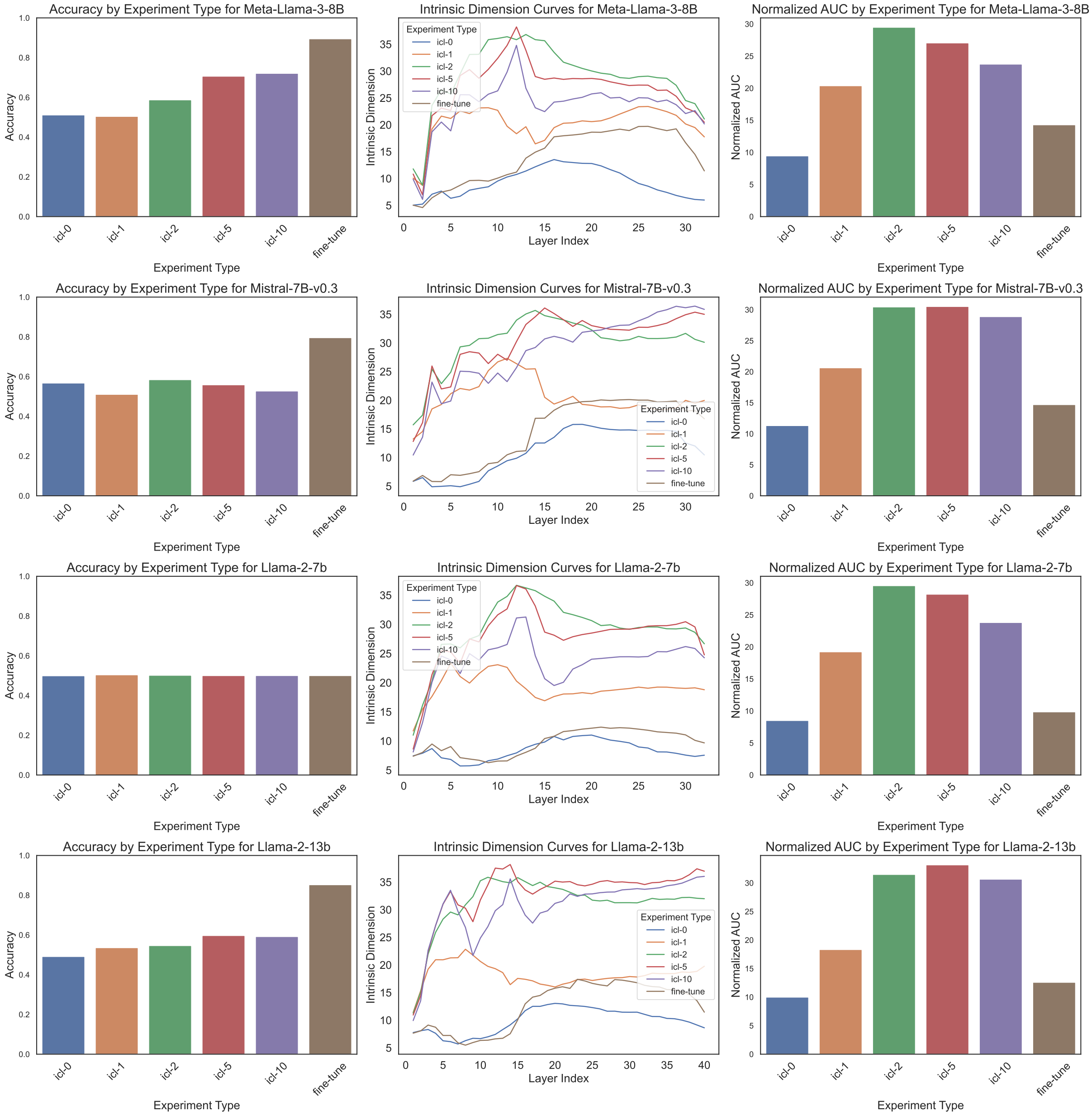


Figure 31: Comparison of Experimental Results for MNLI

Dataset: QNLI



Accuracy by Experiment Type for Meta-Llama-3-8B

Intrinsic Dimension Curves for Meta-Llama-3-8B

Normalized AUC by Experiment Type for Meta-Llama-3-8B

Accuracy by Experiment Type for Mistral-7B-v0.3

Intrinsic Dimension Curves for Mistral-7B-v0.3

Normalized AUC by Experiment Type for Mistral-7B-v0.3

Accuracy by Experiment Type for Llama-2-7b

Intrinsic Dimension Curves for Llama-2-7b

Normalized AUC by Experiment Type for Llama-2-7b

Accuracy by Experiment Type for Llama-2-13b

Intrinsic Dimension Curves for Llama-2-13b

Normalized AUC by Experiment Type for Llama-2-13b

Figure 32: Comparison of Experimental Results for QNLI

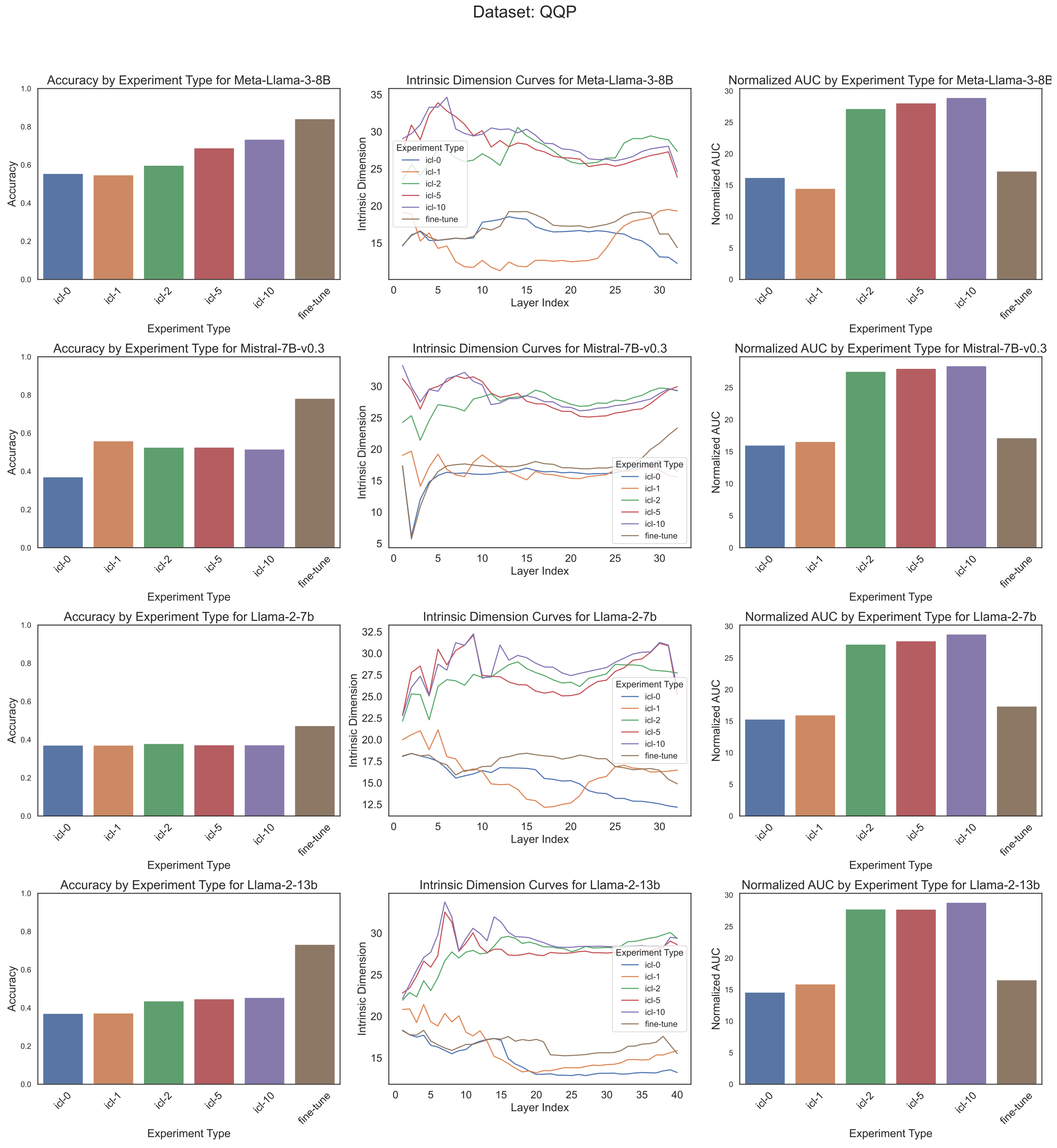


Figure 33: Comparison of Experimental Results for QQP

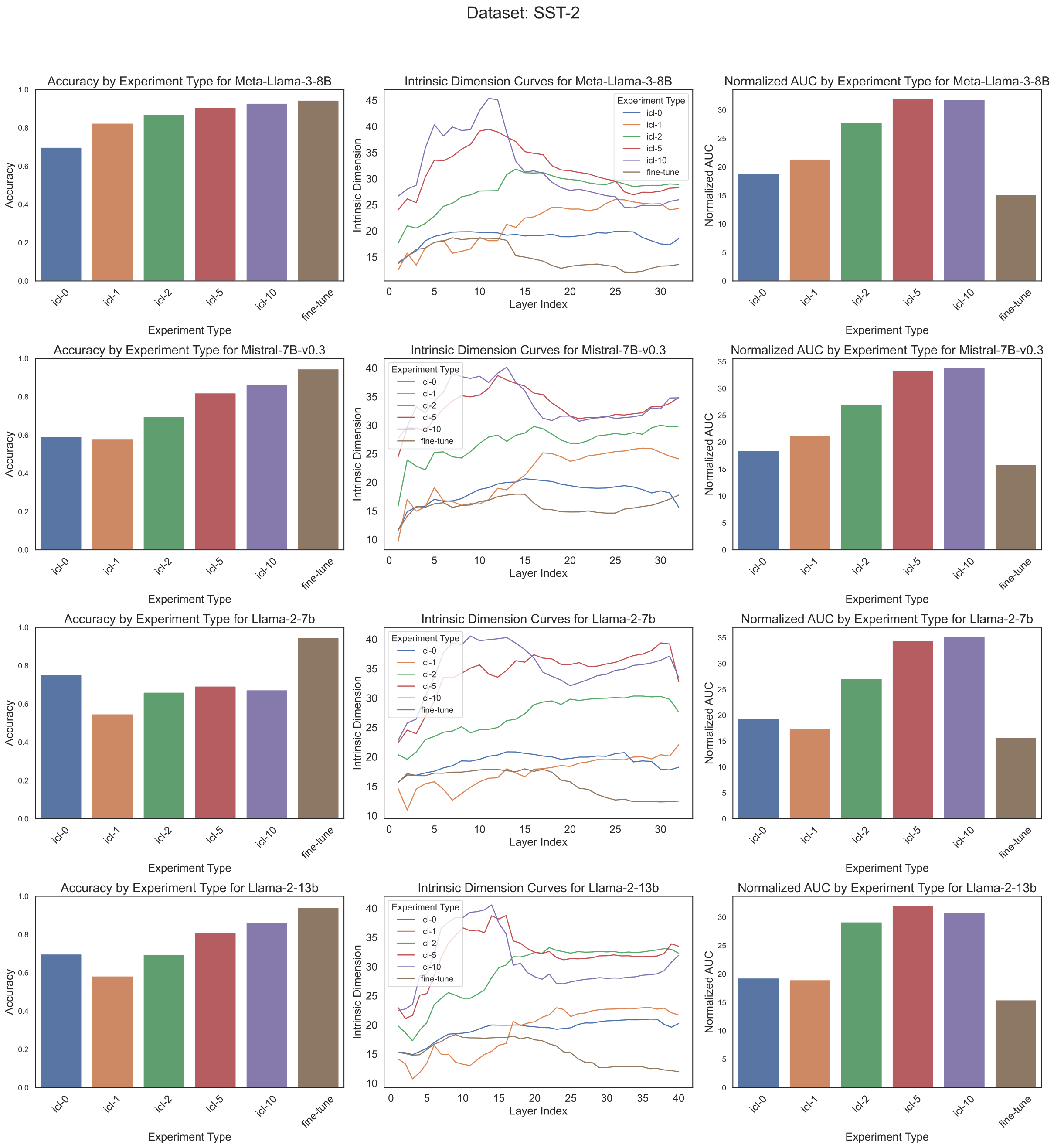


Figure 34: Comparison of Experimental Results for SST-2

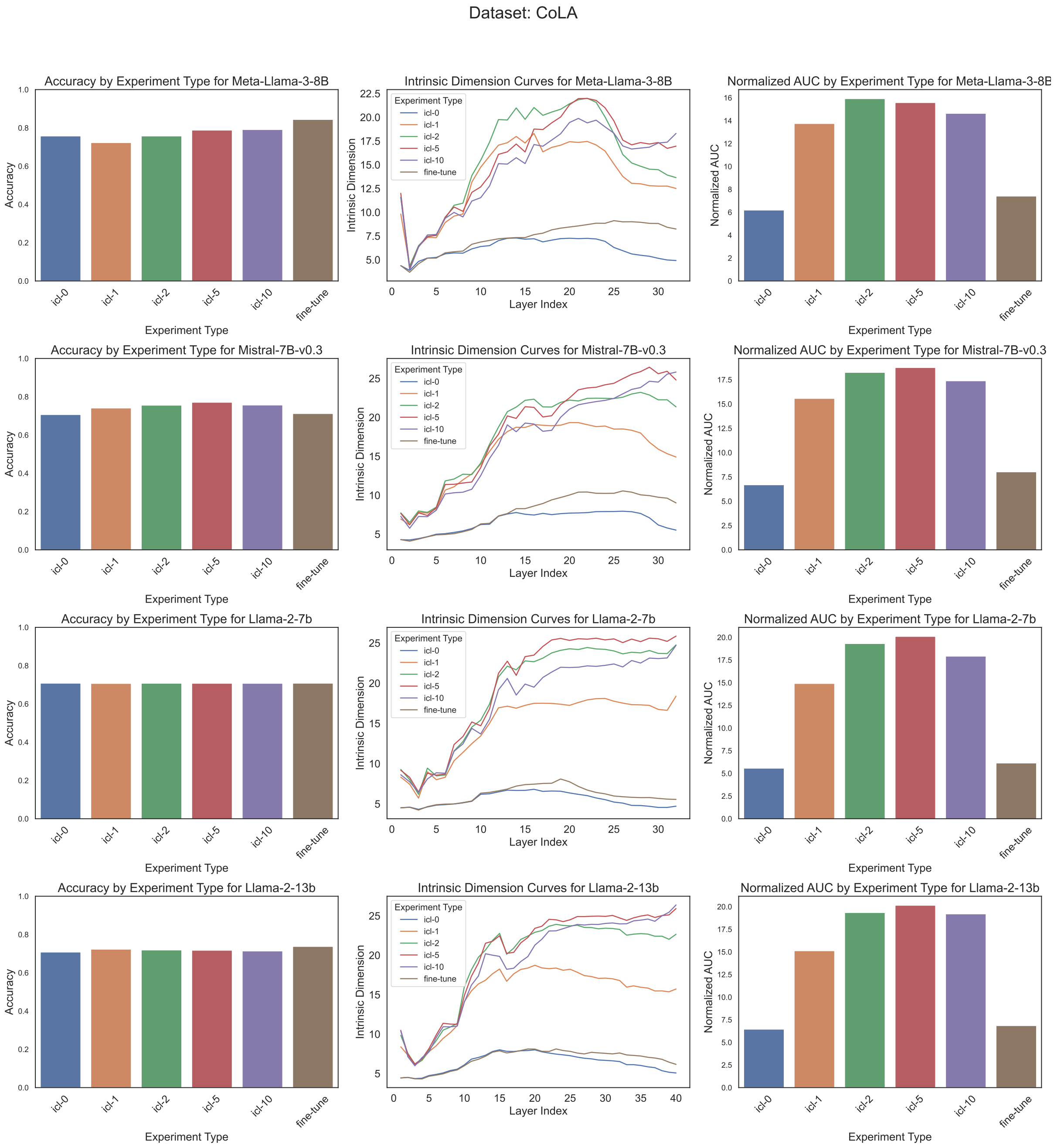
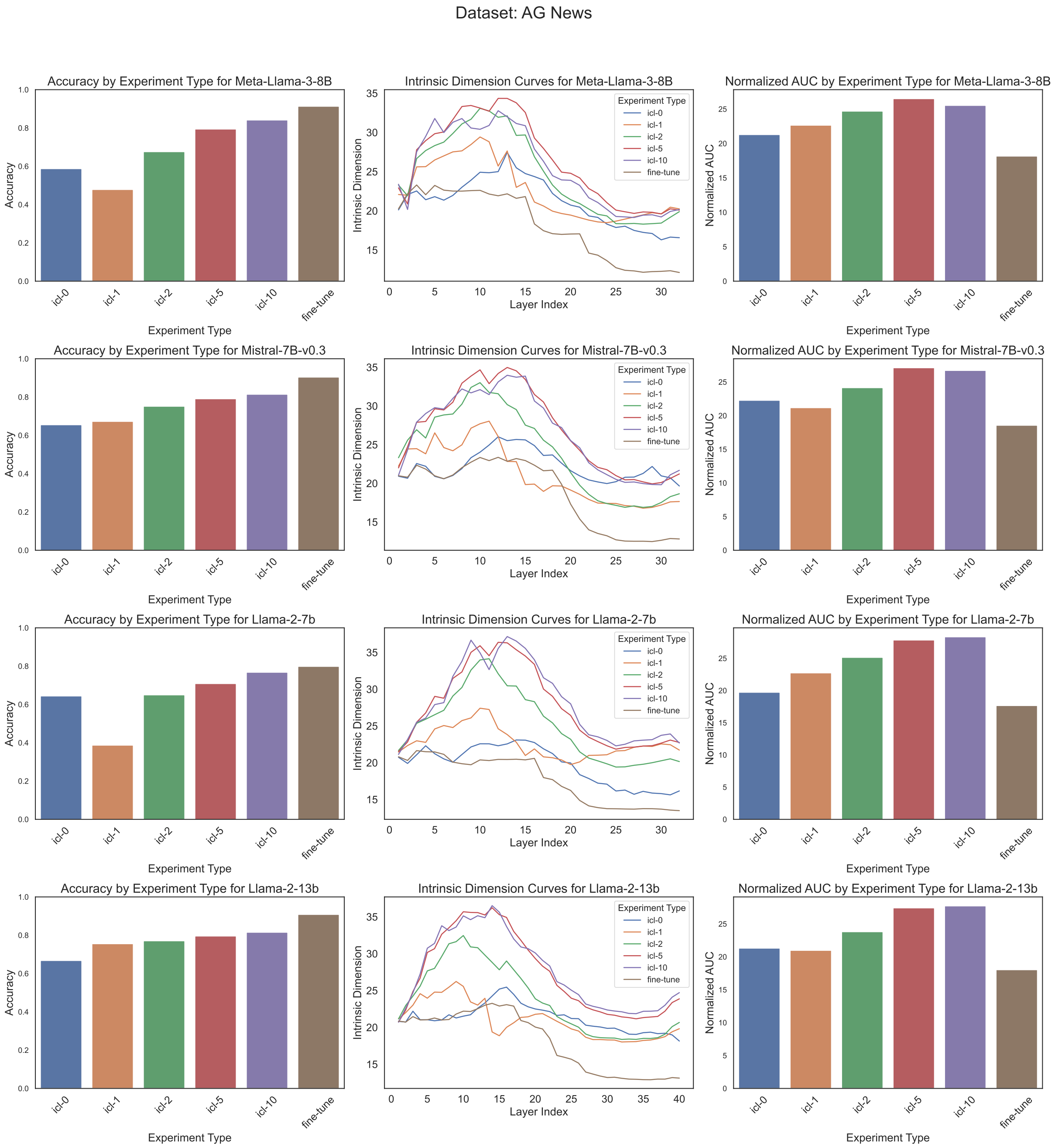


Figure 35: Comparison of Experimental Results for CoLA



Dataset: AG News

Accuracy by Experiment Type for Meta-Llama-3-8B

Intrinsic Dimension Curves for Meta-Llama-3-8B

Normalized AUC by Experiment Type for Meta-Llama-3-8B

Accuracy by Experiment Type for Mistral-7B-v0.3

Intrinsic Dimension Curves for Mistral-7B-v0.3

Normalized AUC by Experiment Type for Mistral-7B-v0.3

Accuracy by Experiment Type for Llama-2-7b

Intrinsic Dimension Curves for Llama-2-7b

Normalized AUC by Experiment Type for Llama-2-7b

Accuracy by Experiment Type for Llama-2-13b

Intrinsic Dimension Curves for Llama-2-13b

Normalized AUC by Experiment Type for Llama-2-13b

Figure 36: Comparison of Experimental Results for AG News

D ICL Experiment Results with Unique Demonstrations

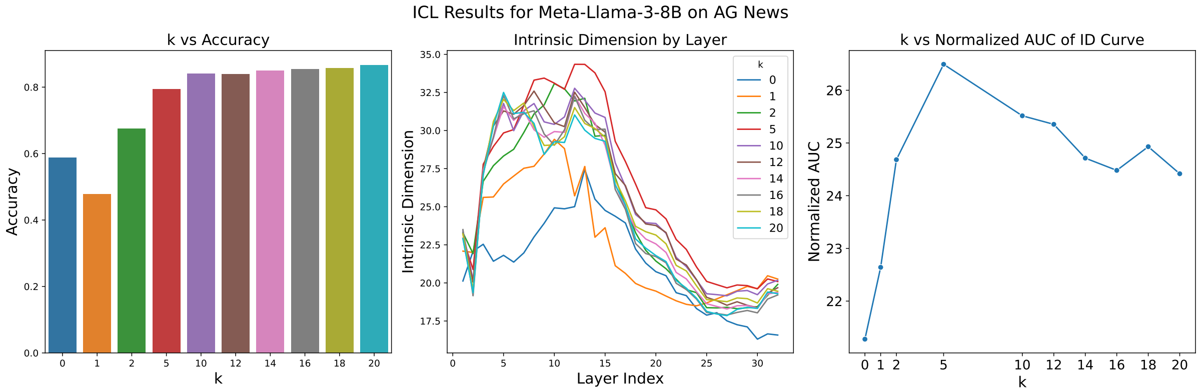


Figure 37: ICL Experiment Results with Unique Demonstrations on AGNews Dataset

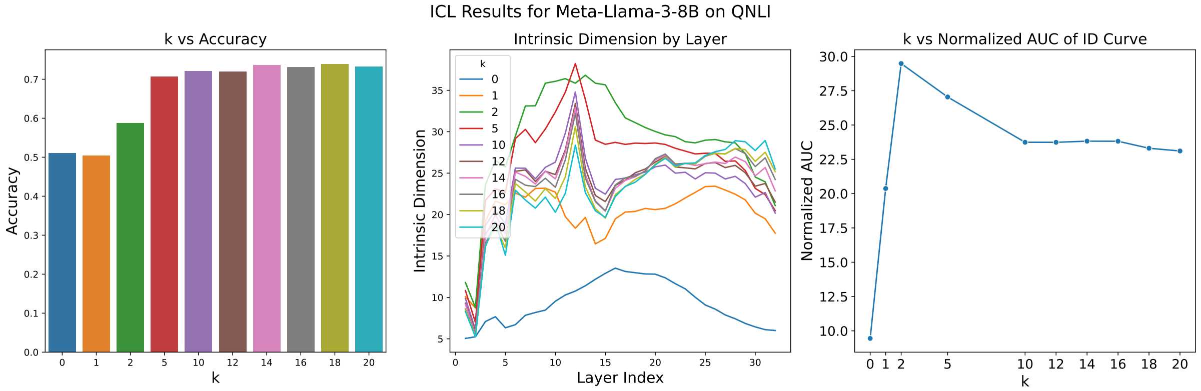


Figure 38: ICL Experiment Results with Unique Demonstrations on QNLI Dataset

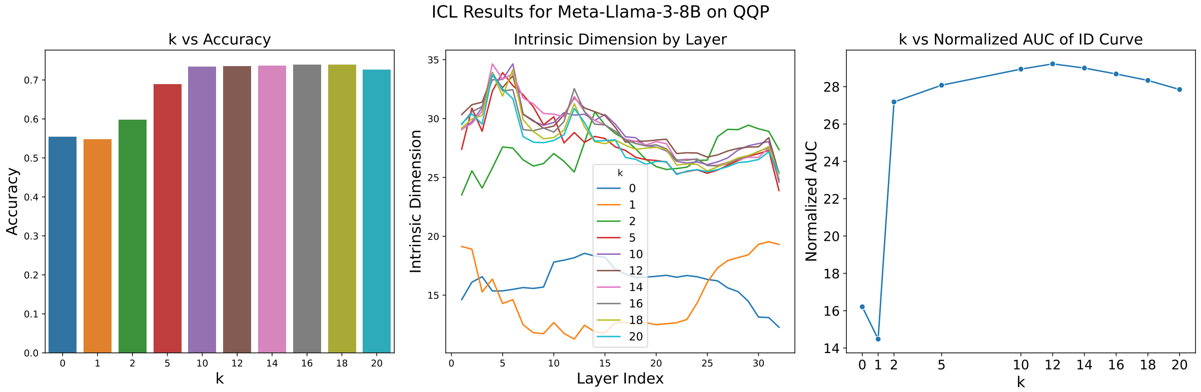


Figure 39: ICL Experiment Results with Unique Demonstrations on QQP Dataset

E Normalized AUC Boxplot by Model for All Learning Paradigms

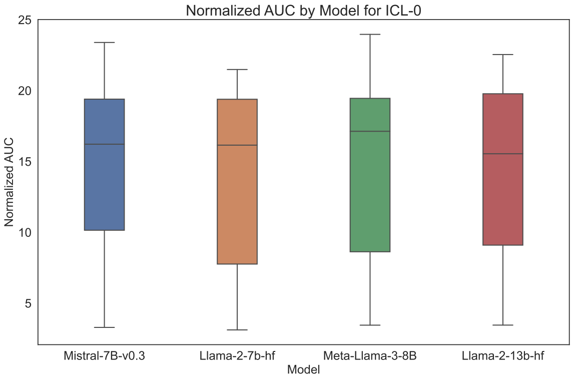


Figure 40: Normalized AUC by Model boxplot for ICL-0 experiments.

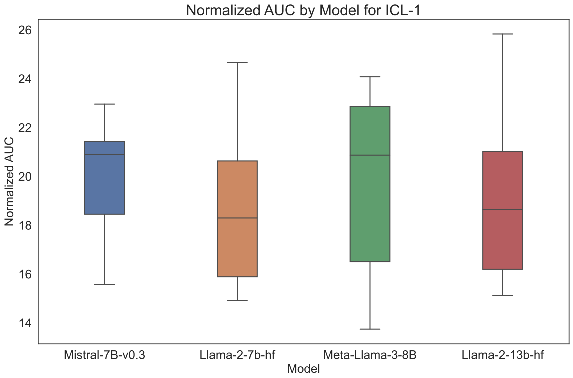


Figure 41: Normalized AUC by Model boxplot for ICL-1 experiments.

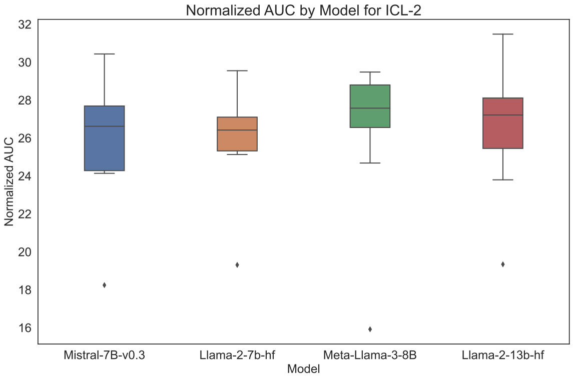


Figure 42: Normalized AUC by Model boxplot for ICL-2 experiments.

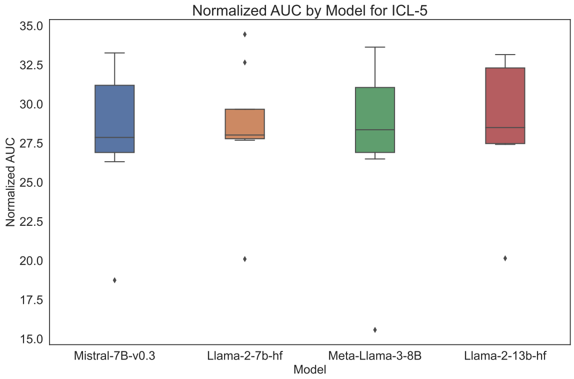


Figure 43: Normalized AUC by Model boxplot for ICL-5 experiments.

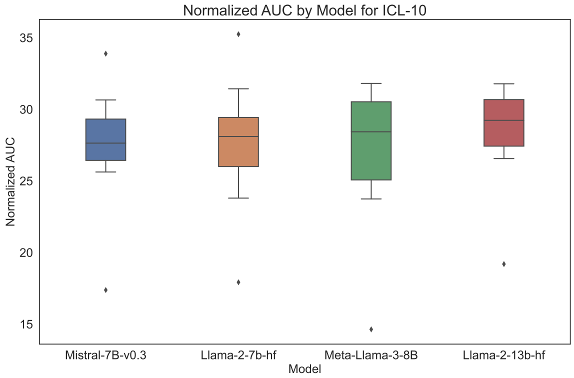


Figure 44: Normalized AUC by Model boxplot for ICL-10 experiments.

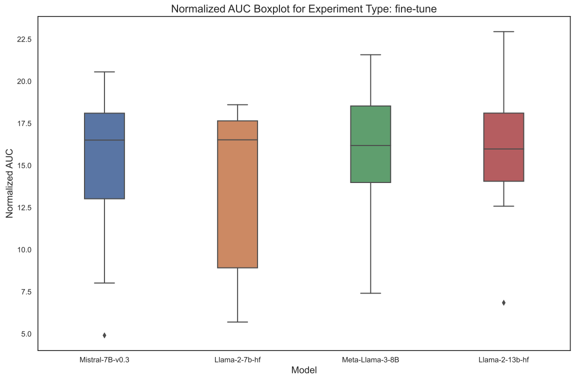


Figure 45: Normalized AUC by Model boxplot for SFT experiments.

F Validating the TwoNN Estimator with the MLE Estimator

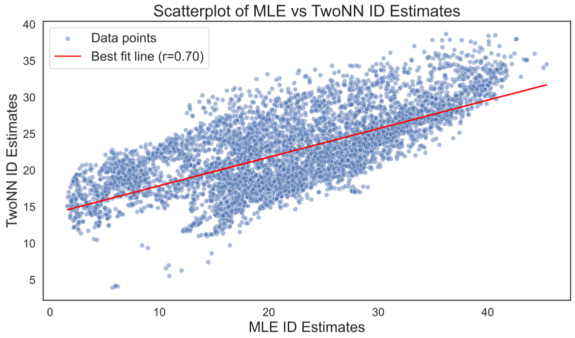


Figure 46: Scatterplot plotting ID estimation results for all experiments using the MLE and TwoNN Estimators.

To assess the validity of our intrinsic dimension estimator, we calculate the intrinsic dimension for different combinations of (learning paradigm, dataset, model, layer) using the TwoNN estimator and Maximum Likelihood Estimator (MLE) introduced by Levina and Bickel (2004). We use a neighborhood of size $k = 50$ when applying MLE. We find that the estimates from the two estimators are correlated with $r = 0.7$. While it is not possible to know the 'true' intrinsic dimensionality of the representations, high correlation between two separate estimators provides a sanity check for our choice of the TwoNN estimator.

G Dataset Generation Details

H Dataset Details

We include details about dataset generation below. We get prompts for all datasets except MMLU from the PromptSource library (Bach et al., 2022).

H.1 QNLI

Items for the training and validation splits in our QNLI experiments were taken from the official QNLI 'train' and 'validation' splits respectively.

Prompt Template:

```
Does that sentence have all you need to
    ↪ answer the question
    ↪ "{{question}}"?
|||
{{answer_choices[label]}}
```

Labels: ['yes', 'no']

H.2 CommonsenseQA

Items for both the training and validation splits in our CommonsenseQA experiments were taken from the official CommonsenseQA 'train' split.

Prompt Template:

```
Given the following options, what do
    ↪ you think is the correct answer
    ↪ to the question below:

{{question}}

Options:
{% for letter, t in zip(answer_choices,
    ↪ choices.text) %}
- {{letter}}: {{t}}
{% endfor %} |||
{{answerKey}}
{% endif %}
```

Labels: ['A', 'B', 'C', 'D']

H.3 MMLU

Items for both the training and validation splits in our MMLU experiments were taken from the official MMLU 'test' split.

Prompt Template:

```
# generate input txt and output txt
letters = ['A', 'B', 'C', 'D']
choices = dataset_element['choices']

input_txt =
    ↪ f"{dataset_element['question']}\n\nA:
    ↪ {choices[0]}\nB:
    ↪ {choices[1]}\nC:
    ↪ {choices[2]}\nD:
    ↪ {choices[3]}\nAnswer:"

output_txt = letters[answer_idx]
combined = input_txt + output_txt
```

H.4 SST-2

Items for both the training and validation splits in our SST-2 experiments were taken from the official SST-2 'train' split.

Prompt Template:

```
{{sentence}}
Question: Was that sentence
    ↪ {{"positive"}} or
    ↪ {{"negative"}}? Answer: ||| {{
    ↪ answer_choices[label] }}
```

Labels: ['negative', 'positive']

H.5 CoLA

Items for both the training and validation splits in our CoLA experiments were taken from the official CoLA 'train' split.

Prompt Template:

```
Does the following sentence make sense
    ↪ and use correct English? Please
    ↪ answer {{"yes"}} or {{"no"}}.
{{sentence}}
|||
{{ answer_choices[label] }}
```

Labels: ['no', 'yes']

H.6 AGNews

Items for the training and validation splits in our AGNews experiments were taken from the official AGNews 'train' and 'validation' splits respectively.

Prompt Template:

```
What label best describes this news
    ↪ article?
{{text}} |||
{{answer_choices[label] }}
```

Labels: ['World politics', 'Sports', 'Business', 'Science and technology']

H.7 MNLI

Items for the training and validation splits in our MNLI experiments were taken from the official MNLI 'train' and 'validation_matched' splits respectively.

Prompt Template:

```
{{premise}} Are we justified in saying
    ↪ that "{{hypothesis}}"? Yes, no,
    ↪ or maybe? ||| {{
    ↪ answer_choices[label] }}
```

Labels: ['Yes', 'Maybe', 'No']

H.8 QQP

Items for the training and validation splits in our QQP experiments were taken from the official QQP 'train' and 'validation' splits respectively.

Prompt Template:

```
I'm an administrator on the website
    ↪ Quora. There are two posts, one
    ↪ that asks "{{question1}}" and
    ↪ another that asks
    ↪ "{{question2}}". I can merge
    ↪ questions if they are asking the
    ↪ same thing. Can I merge these
    ↪ two questions? ||| {{
    ↪ answer_choices[label] }}
```

Labels: ['no', 'yes']