Layer by Layer: Uncovering Where Multi-Task Learning Happens in Instruction-Tuned Large Language Models

Zheng Zhao[1] **Yftah Ziser**[2] **Shay B. Cohen**[1]
[1]Institute for Language, Cognition and Computation, University of Edinburgh
[2]Nvidia Research
zheng.zhao@ed.ac.uk , yziser@nvidia.com , scohen@inf.ed.ac.uk

Abstract

Fine-tuning pre-trained large language models (LLMs) on a diverse array of tasks has become a common approach for building models that can solve various natural language processing (NLP) tasks. However, where and to what extent these models retain task-specific knowledge remains largely unexplored. This study investigates the task-specific information encoded in pre-trained LLMs and the effects of instruction tuning on their representations across a diverse set of over 60 NLP tasks. We use a set of matrix analysis tools to examine the differences between the way pre-trained and instruction-tuned LLMs store task-specific information. Our findings reveal that while some tasks are already encoded within the pre-trained LLMs, others greatly benefit from instruction tuning. Additionally, we pinpointed the layers in which the model transitions from high-level general representations to more task-oriented representations. This finding extends our understanding of the governing mechanisms of LLMs and facilitates future research in the fields of parameter-efficient transfer learning and multi-task learning.[1]

1 Introduction

While pre-trained LLMs exhibit impressive performance across diverse tasks and demonstrate remarkable generalization capabilities (Brown et al., 2020; Wei et al., 2022b; Touvron et al., 2023; Chowdhery et al., 2023; OpenAI et al., 2024), the representations they learn and the task-specific information encoded during pre-training remain largely opaque and unexplored.

Recent research has investigated fine-tuning strategies to adapt LLMs to specific tasks, including supervised fine-tuning on task-specific datasets and instruction tuning (Mishra et al., 2022; Chung et al., 2022; Sanh et al., 2022). While these approaches have shown promising results in tailoring LLMs for improved task performance, a comprehensive understanding of their impact on the learned representations is still lacking.

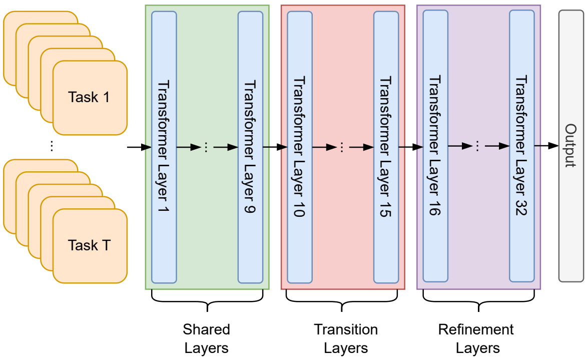


Figure 1: An illustration of our findings using the Llama 2 7B model (Touvron et al., 2023) as an example. We show that when instruction tuning on T different tasks, the layers are divided into three functional sections: the shared layers (layers 1 to 9) form general representations shared among all tasks, the transition layers (layers 10 to 15) transition the representations into task-specific information, and the refinement layers (layers 16 to 32) continue to refine the representations toward specific tasks.

In this study, we perform a set of analyses to investigate task-specific information encoded in pre-trained LLMs and the effects of instruction tuning on their representations. The analysis leverages a sub-population analysis technique called Model-Oriented Sub-population and Spectral Analysis (MOSSA; Zhao et al. 2022), which provides an alternative to traditional probing methods for analyzing model representations within specific sub-populations of the training data. MOSSA involves comparing two models: a *control* model trained on the data relevant to the sub-population of interest (e.g., a particular task), and an *experimental* model that is identical to the control model but is also trained on additional data from different

[1] Our code is available at: https://github.com/zsquaredz/layer_by_layer/

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 15195–15214
November 12-16, 2024 ©2024 Association for Computational Linguistics

sources (e.g., multiple tasks). By analyzing the representational differences between these models, we can isolate the task-specific information encoded within the control model for the sub-population of interest.

To compare the representations learned by different LLM variants, we leverage the Center Kernel Alignment (CKA; Kornblith et al., 2019) metric. CKA measures the alignment between representations in a kernel space, providing a robust measure of similarity that is insensitive to scaling and centering. By using MOSSA and CKA, we investigate the following research questions:

1. To what extent are different NLP tasks already encoded in pre-trained LLMs?
2. In what ways does instruction tuning modify the representational landscape of LLMs?
3. Do the representational effects of instruction tuning generalize to unseen tasks?

Through an extensive analysis spanning over 60 diverse NLP tasks following the Flan framework (Longpre et al., 2023), we shed light on the underlying mechanisms that govern the encoding and adaptation of task-specific information within LLMs under instruction tuning. A key finding of our work is the identification of three functional groups of layers: a) shared layers, in which more general information is learned and shared across tasks; b) transition layers, in which task-specific information is intensified; c) refinement layers, in which the LLMs continue to refine their representations towards task-specific predictions. Our findings contribute to a deeper understanding of the inner workings of LLMs and hold promising implications for future research in parameter-efficient fine-tuning (PEFT), multi-task learning (MTL), and model compression, benefiting a wide range of NLP applications.

We structure this study as follows: §2 describes our methodology for our analysis, while §3 outlines the experimental setup and tools used to train and analyze our LLMs. §4 then attempts to answer each of the research questions outlined above by presenting and analyzing our results. Finally, in §5, we summarize our key findings and discuss their potential implications.

2 Methodology

We use the MOSSA framework introduced by Zhao et al. (2022). Unlike standard probing methods (Belinkov et al., 2017a,b; Giulianelli et al., 2018), which build a model to predict a downstream task for quantifying encoded information, MOSSA compares representations from two models: a control model trained on data of interest and an experimental model trained on additional data from different sources. Here, the data of interest refers to tasks. Probing methods, while useful, can be limited because they rely on different metrics to evaluate performance across various tasks, making it challenging to directly compare the amount of information stored about tasks as diverse as sentiment analysis and translation. MOSSA, on the other hand, circumvents this issue by comparing the latent representations of models rather than their downstream performance metrics. MOSSA calculates the similarity between the representations of the control and experimental models, thus representing the information captured from the relevant sub-population of data through their latent representations. By comparing different models to each other, we can learn what information is captured when a subset of the data is used versus the whole dataset.

We use matrix analysis to compare representation similarity between the experimental model, such as pre-trained, instruction-tuned, and corresponding single-task control models trained on individual tasks. Intuitively, a high similarity between the experimental and control models indicates the experimental model stores task-specific information learned by the control model, which was fine-tuned solely on data from that task. The similarity is measured using the CKA metric, which quantifies the similarity between two representations in a kernel space.

Formally, let $[T]$ be an index set of tasks, and let $\mathbf{E}$ be the experimental model and $\mathbf{C}_t$ be the control model for task $t \in [T]$. We assume a set of inputs $\mathcal{X} = \bigcup_{t=1}^{T} \mathcal{X}_t$, where each $\mathcal{X}_t = \{\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,n}\}$ represents a set of input instructions for task t. For simplicity, we assume that all sets have the same size n, although this is not a strict requirement.[2]

For each $t \in [T]$ and $i \in [n]$, we apply the experimental model $\mathbf{E}$ and the control model $\mathbf{C}_t$ to the input instruction $\mathbf{x}_{t,i}$ to obtain two corresponding representations $\mathbf{y}_{t,i} \in \mathbb{R}^d$ and $\mathbf{z}_{t,i} \in \mathbb{R}^{d_t}$, respectively. Here, d is the dimension of the experimental model representations, and d_t is the dimension of

[2] In our actual experimental setup for this work, we use different dataset sizes for each task, which reflects real-world scenarios. For more details, please refer to §3.

the control model representations for task t. To obtain the representations $\mathbf{y}_{t,i}$ and $\mathbf{z}_{t,i}$, we use the last token representation following previous work (Qiu et al., 2024; Wang et al., 2024), as LLMs are decoder-only and the last token captures all input information. These representations can be extracted from any layers of the respective models.

By stacking these vectors into two matrices for each task t, we obtain the paired matrices $\mathbf{Y}_t \in \mathbb{R}^{n\times d}$ and $\mathbf{Z}_t \in \mathbb{R}^{n\times d_t}$. We calculate the CKA value between $\mathbf{Y}_t$ and $\mathbf{Z}_t$ following the procedure:

- Computing the kernel matrices $K_{\mathbf{Y}_t} \in \mathbb{R}^{n\times n}$ and $K_{\mathbf{Z}_t} \in \mathbb{R}^{n\times n}$ for $\mathbf{Y}_t$ and $\mathbf{Z}_t$, respectively, using the same kernel function (e.g., linear, Gaussian, or polynomial).[3]
- Centering the kernel matrices by $K_{\mathbf{Y}_t} = K_{\mathbf{Y}_t} - \frac{1}{n}\mathbf{1}K_{\mathbf{Y}_t} - \frac{1}{n}K_{\mathbf{Y}_t}\mathbf{1} + \frac{1}{n^2}\mathbf{1}K_{\mathbf{Y}_t}\mathbf{1}$, similarly for $K_{\mathbf{Z}_t}$, where $\mathbf{1}$ is a matrix of ones.
- Computing the CKA value by first compute the Frobenius inner product of the centered Gram matrices: $\mathrm{HSIC}(K_{\mathbf{Y}_t}, K_{\mathbf{Z}_t}) = \mathrm{Tr}(K_{\mathbf{Y}_t}^\top K_{\mathbf{Z}_t})$, where Tr denotes the trace of a matrix. Then normalize the CKA value:

$$\mathrm{CKA}(\mathbf{Y}_t,\mathbf{Z}_t)=\frac{\mathrm{HSIC}(K_{\mathbf{Y}_t},K_{\mathbf{Z}_t})}{\sqrt{\mathrm{HSIC}(K_{\mathbf{Y}_t},K_{\mathbf{Y}_t})\cdot\mathrm{HSIC}(K_{\mathbf{Z}_t},K_{\mathbf{Z}_t})}}. \quad (1)$$

While other similarity metrics like SVCCA (Raghu et al., 2017) exist, they have a limitation due to the constraint of being invariant to invertible linear transformations, which requires the number of data points to be greater than the number of representation dimensions. We use CKA as it has shown robust results when the data sample is smaller (Kornblith et al., 2019), as is sometimes the case for datasets used in our work.

Our method provides an approach to quantify the task-specific information encoded in the representations of LLMs. By comparing the experimental model's representations with those of single-task control models, we can gain insights into the extent to which the experimental model captures task-specific knowledge and how this knowledge is distributed across its representations.

3 Experimental Setup

Data We use the Flan 2021 dataset (Wei et al., 2022a) to fine-tune our LLMs. The Flan dataset is a comprehensive collection of more than 60 NLP datasets, including both language understanding and generation tasks. These datasets are organized into twelve task clusters, where datasets within a given cluster belong to the same task type. To enhance instruction diversity, we follow the approach of Wei et al. (2022a) and use ten unique natural language instruction templates for each dataset. These templates provide varying descriptions of the task to be performed. Our instruction tuning pipeline combines all datasets and randomly samples from each dataset during training. To mitigate the impact of dataset size imbalances, we limit the number of training examples per task cluster to 50k and use the examples-proportional mixing scheme (Raffel et al., 2020) with a mixing rate maximum of 3,000 per task. This means that no task receives additional sampling weight for examples in excess of 3,000. We provide further details about the dataset in Appendix A.

Models We have two types of models: the experimental model $\mathbf{E}$, fine-tuned using all T available tasks, and the single-task model $\mathbf{C}_t$ for $t \in [T]$, fine-tuned only on the t-th task. In some experiments, the model $\mathbf{E}$ can also be the pre-trained model. We use the Llama 2 models (Touvron et al., 2023) as the starting training checkpoint for both $\mathbf{E}$ and $\mathbf{C}_t$. Specifically, we use the 7B variant, which consists of 32 layers and 4096 hidden dimensions. This model allows us to conduct a more comprehensive set of experiments while maintaining control over experimental conditions. Since we have over 60 control models, exploring larger models or different families would have been computationally infeasible due to resource constraints. Given these limitations, we choose to fully explore a realistic multi-task scenario, involving more than 60 different tasks, with the aim of extracting significant findings that we expect to generalize to other models.

Training We use LoRA (Hu et al., 2022) for fine-tuning our LLMs, with the rank r set to 8. We use the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of 5×10^{-5} for fine-tuning the instruction dataset. We use the same vocabulary, tokenizer, and learning rate scheduler for Llama 2-7B as in Touvron et al. (2023). We train the multi-task model $\mathbf{E}$ (which we refer to as Llama 2-SFT in our experiment) for a maximum of 100K steps and the single-task models $\mathbf{C}_t$ for a maximum of 10K steps, using validation set cross-entropy loss for early stopping. Our multi-task models are

[3] For linear kernel, which is what we use in our experiment, $K_{\mathbf{Y}_t} = \mathbf{Y}_t\mathbf{Y}_t^\top$, and $K_{\mathbf{Z}_t} = \mathbf{Z}_t\mathbf{Z}_t^\top$.

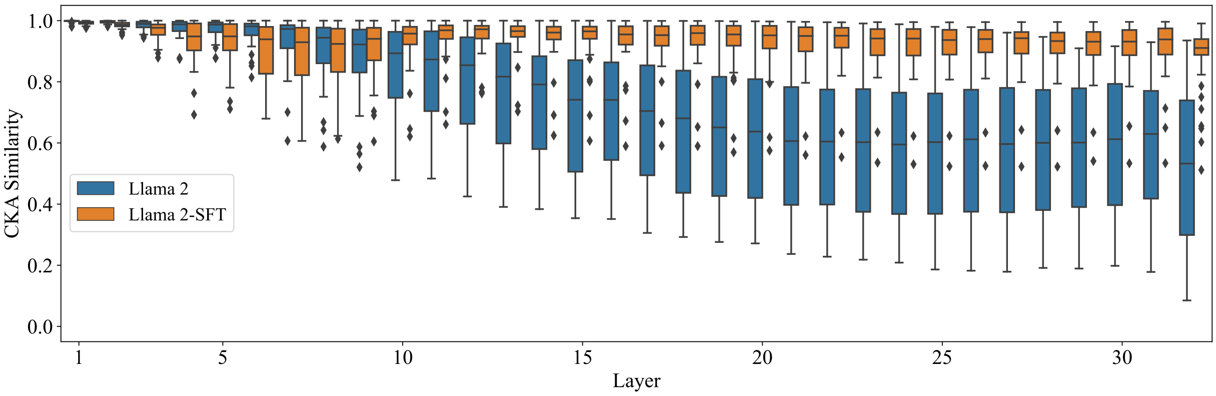


Figure 2: Distribution of CKA similarities across all layers for the pre-trained Llama 2 model and the instruction-tuned Llama 2-SFT model. The boxplots illustrate the spread and variation of CKA similarities between each model and the control models across different tasks. The comparison between the two models highlights the impact of instruction tuning on shaping task-specific representations in different layers.

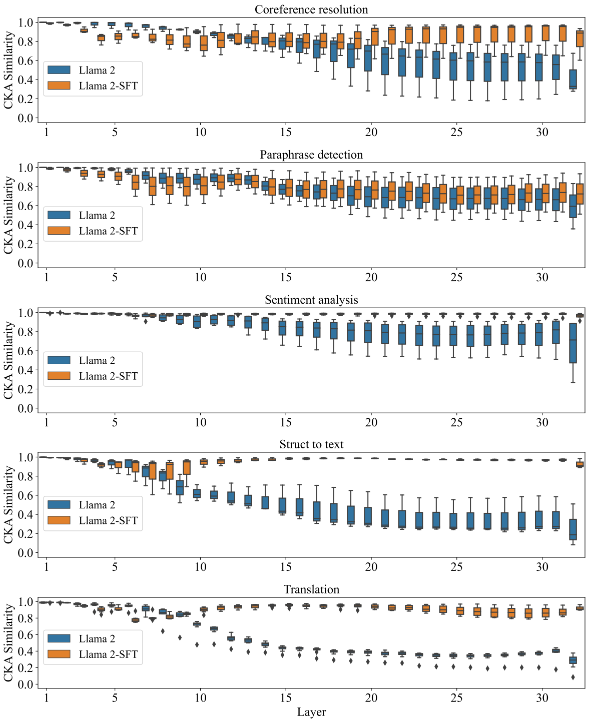


Figure 3: Distribution of CKA similarities across all layers for the pre-trained Llama 2 model and the instruction-tuned Llama 2-SFT model, grouped by different task clusters.

trained on four NVIDIA A100 GPUs with a batch size of 16 per GPU, while single-task models are trained on one NVIDIA A100 GPU with a batch size of 16.We use PyTorch (Paszke et al., 2019), the HuggingFace library (Wolf et al., 2020), and the LLaMA-Factory library (Zheng et al., 2024) for all model implementations and LoRA fine-tuning.

4 Experiments and Results

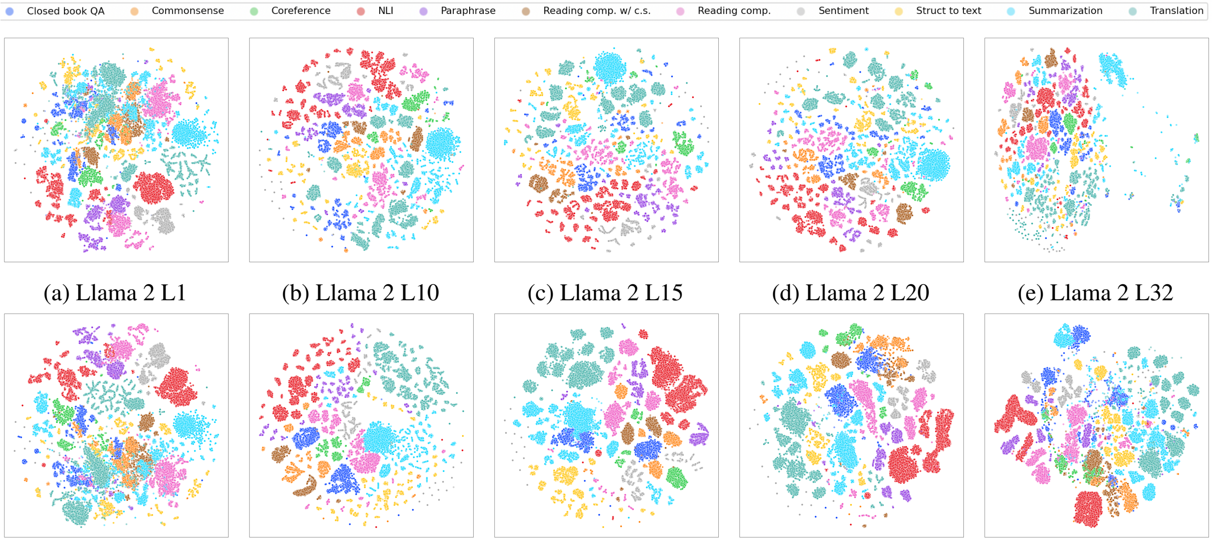
To shed light on the underlying mechanisms of MTL (Caruana, 1997) in LLMs, we start by examining what NLP tasks are encoded in the pre-trained LLM representations, establishing a baseline for comparison with the instruction-tuned model (§4.1). Then, using matrix analysis methods, we contrast the representational properties of the pre-trained and instruction-tuned LLMs to understand the effects of instruction tuning (§4.2, 4.3, and 4.4). Finally, we evaluate the generalization of our findings to unseen tasks (§4.5).

4.1 Task Information in Pre-trained LLMs

To identify task-relevant information in pre-trained LLMs, we compared representations from the pre-trained Llama 2 model with task-specific fine-tuned models ($\{\mathbf{C}_t\}_t$). Figure 2 shows the distribution of CKA similarities across all tasks and layers for the Llama 2 model. The CKA similarities between pre-trained Llama 2 and control models generally decrease through higher layers.

Llama 2 maintains high CKA similarities in earlier layers, and since CKA compares against control models fine-tuned on individual tasks, this suggests that representational changes in the earlier layers are minimal across tasks. However, we observe widespread variance in CKA values across different tasks in the middle and higher layers, suggesting that some tasks are better captured in the Llama 2 model representations than others.

To gain a more fine-grained understanding, we analyzed the CKA results at the task cluster level, where each cluster consists of a group of similar tasks. The Flan dataset organizes tasks into 12 dif-



(a) Llama 2 L1 (b) Llama 2 L10 (c) Llama 2 L15 (d) Llama 2 L20 (e) Llama 2 L32

(f) Llama 2-SFT L1 (g) Llama 2-SFT L10 (h) Llama 2-SFT L15 (i) Llama 2-SFT L20 (j) Llama 2-SFT L32

Figure 4: t-SNE visualizations of the representations for each task cluster in different layers of the pre-trained Llama 2 model and the instruction-tuned Llama 2-SFT model. Each subplot presents the t-SNE projection of the representations, color-coded by task cluster, for a specific layer of the respective model. “Reading comp.” denotes reading comprehension tasks, and “reading comp. w/ c.s.” denotes reading comprehension tasks with commonsense reasoning.

ferent clusters, detailed in Appendix A. We present CKA results for a selection of representative clusters in Figures 3, with the full results provided in Appendix B.2.

For clusters like closed-book QA, commonsense reasoning, paraphrase detection, and sentiment analysis, which heavily rely on general linguistic and semantic understanding, the CKA similarity for Llama 2 is high. This indicates that pre-trained models already encode these tasks well in their representations. Conversely, for clusters like coreference resolution, reading comprehension, structured data to text generation, summarization, and translation, which require specialized, structured, or domain-specific knowledge involving complex transformations or extended context management, the CKA similarities are low, suggesting that next token prediction at pre-training is insufficient for encoding these tasks.

4.2 Impact of Instruction Tuning

Mapping Layers to Their Functionality To investigate how instruction tuning affects the representations learned by LLMs, we compared the instruction-tuned model (Llama 2-SFT) with task-specific fine-tuned control models. As illustrated in Figure 2, the CKA similarities between Llama 2-SFT and the control models do not decrease as significantly as those for the pre-trained model (Llama 2) across layers. In the early layers (1 to 9), we observe that for many tasks, the CKA scores are lower for Llama 2-SFT compared to Llama 2, indicating that Llama 2-SFT representations diverge from those of the control models, which were fine-tuned on individual tasks (thus specializing in them). This suggests that, unlike the Llama 2 model, training Llama 2-SFT on a high number of tasks encourages it diverge from the control models' representations and learn more general representations in the lower layers, a characteristic typical of MTL models. We denote layers 1-9 as “shared layers”, as our findings suggest their representations are shared across tasks, similar to more studied MTL models.

In the middle layers (10-15), there is a significant transition, with the Llama 2-SFT model exhibiting high similarity to *all control models*. This indicates that these layers encode a high degree of task-specific information, as their representations are almost identical to those of the specialized control models. We denote layers 10-15 as “transitional layers”, as our findings suggest the transition to task-specific representations occurs within these layers. This trend continues, albeit to a lesser extent, up to the final layers (16-32), which we denote as “refinement layers”, as they keep refining the representations up to the final prediction. Based on our findings, we can map each layer in the Llama 2-SFT model to its corresponding function with re-

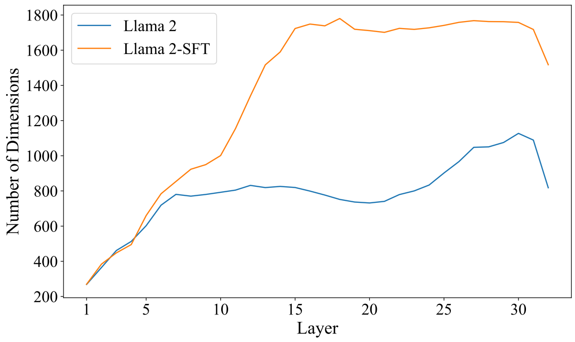


Figure 5: Average number of dimensions required to explain 99% of the representational variance across all tasks, as a function of the layer number.

spect to MTL (see Figure 1). While previous work (Wei et al., 2022a; Chung et al., 2022) has empirically demonstrated the effectiveness of instruction tuning for improving performance on a variety of NLP tasks, to the best of our knowledge, we are the first to propose such a mapping. In the following sections, we provide additional analyses to further validate our mapping.

Examining individual task clusters Figures 3 demonstrates that for tasks that are not well encoded in the pre-trained Llama 2 (e.g., structured data to text generation, translation), the CKA similarities from the instruction-tuned Llama 2-SFT remained high throughout all transition and refinement layers (10-32). Instruction tuning for these tasks induced significant representational shifts, adapting the model's internal structure to meet their specific demands. This aligns with prior work (Aghajanyan et al., 2021) showing that tasks requiring more sophisticated reasoning and modeling benefit greatly from task-specific tuning of pre-trained language models.

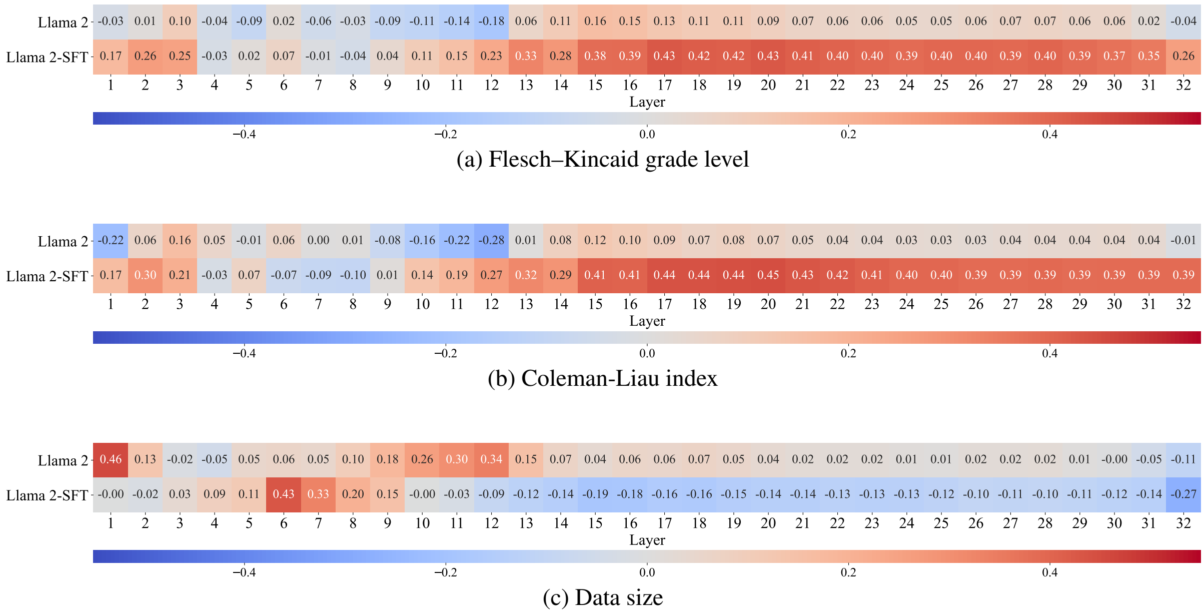
4.3 Representation Clustering and Variance Analysis

To further investigate representational differences, we used t-SNE (Van der Maaten and Hinton, 2008) to visualize task clusters across layers. Figure 4 presents a representative selection of layers, including a shared layer (layer 1), transition layers (layers 10 and 15), and refinement layers (layers 20 and 32). The full results for all layers are provided in Appendix B.2. In the first layer, both Llama 2 and Llama 2-SFT exhibit similar clustering. However, as we move to the transition layers, from layers 10 to 15, the Llama 2-SFT model forms more distinct task clusters compared to the Llama 2 model. This is further evidence that instruction tuning transforms the representations towards task-specificity in the transition layers. This clustering becomes increasingly pronounced in refinement layers, highlighting the effectiveness of instruction tuning in differentiating task-specific information and enhancing the ability to specialize representations for different tasks.

To quantify these differences, we performed variance analysis on the representations. We sought to determine if the model's ability to retain a large amount of task-specific information for many tasks affects its representation complexity. We analyzed the number of principal components required to explain 99% of the variance in representation matrices across layers. The average number of components over all tasks is presented in Figure 5. In the shared layers, both Llama 2 and Llama 2-SFT models require a similar number of dimensions. Then, in the transition layers, Llama 2-SFT model begins to require more dimensions, suggesting it captures more complex task-specific information. This further demonstrates that the transition layers are indeed the layers where the transition to the task-specific representations occurs.

4.4 Assessing Task Specific Information via Readability

In the preceding sections, we observed that the Llama 2 model exhibited a high variance in the amount of task-specific information stored across different tasks. In contrast, the Llama 2-SFT model demonstrated a low variance, storing a high level of task-specific information in its transition and refinement layers. While the Llama 2-SFT model exhibited low variance, we aimed to investigate the task priorities within the representation and identify features that could predict it. Previous research by Zhao et al. (2022) has shown that when masked language models, such as BERT (Devlin et al., 2019), are trained on data from multiple domains, they tend to allocate their parameters to store domain-specific information. Unlike our approach, which examines instruction-level representations using the last token of an instruction, their study used the MOSSA method to analyze contextualized word embeddings, allowing them to focus on domain-specific words. We followed a similar analysis to examine task-specific information, which is strongly related to domain-specific information (as tasks can be viewed as domains). We used readability as a proxy for domain-specific information,



(a) Flesch–Kincaid grade level

(b) Coleman-Liau index

(c) Data size

Figure 6: Pearson correlation results between the CKA similarities for all tasks, their reading difficulty, and data size across all layers. Higher values in reading difficulty measures correspond to greater reading difficulty.

relying on the finding by Pitler and Nenkova (2008) that texts with more domain-specific and less commonly used words tend to have lower readability, resulting in higher reading difficulty scores.

We used two highly popular reading difficulty measures: the Flesch-Kincaid grade level score (Kincaid et al., 1975) and the Coleman-Liau Index (Coleman and Liau, 1975). The Flesch-Kincaid score assesses text readability based on factors like average sentence length and syllables per word, with lower scores indicating easier reading. Similarly, the Coleman-Liau Index estimates the required reading grade level based on characters, words, and sentences, with higher values corresponding to greater difficulty. We performed Pearson correlation analyses between CKA similarity and reading difficulty measures for all tasks across all layers. Specifically, we first calculated the readability measure for each input instruction, then obtained CKA similarities for representations from each layer. Finally, we computed the Pearson correlation coefficients between each input's readability measure and the corresponding CKA similarities from each layer.

As illustrated in Figure 6a, we found a positive correlation between CKA similarity and the Flesch-Kincaid score for Llama 2-SFT. This correlation rapidly increases between layer 10 and layer 15 (the transition layers) and then saturates. These transitional layers are where task specialization transformations occur, as discussed earlier. This correlation is much weaker for the Llama 2 model. A similar pattern is observed with the Coleman-Liau Index, as shown in Figure 6b. These findings suggest that instruction-tuned models encode more information for tasks with more task-specific vocabulary, as measured by their texts' readability indices. These findings thus suggest that instruction-tuned models encode and preserve task-specific information in the transition layers and retain it through the refinement layers, complementing our earlier findings. Moreover, we previously noted that one of the advantages of CKA, compared to other similarity metrics, is its minimal requirement for a large number of data points in the analysis. To verify this, we conducted a correlation analysis between data size and CKA similarity, with the results presented in Figure 6c. The analysis revealed no clear correlation between data size and CKA similarities, indicating that the number of data points used for CKA per task does not impact the CKA similarity.

4.5 Evaluating Representations on Unseen Tasks

While our previous analyses focused on evaluating representations against models trained on the same task data, it is crucial to examine how well our findings generalize to unseen tasks. To investigate this, we held out a set of seven tasks, including conversational question answering, question classification, math problems, linguistic acceptability, and word sense disambiguation (details in Appendix A). Our

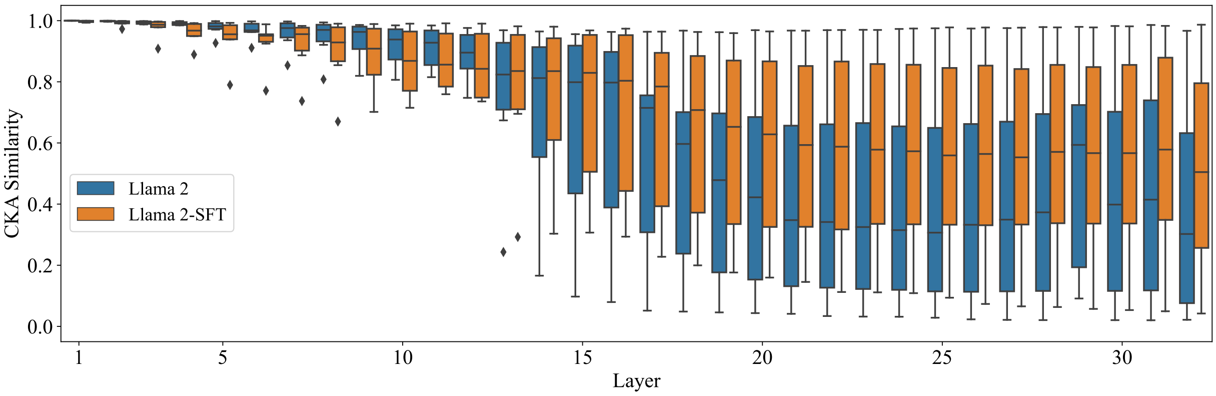


Figure 7: Distribution of CKA similarities across all layers for the pre-trained Llama 2 model and the instruction-tuned Llama 2-SFT model on unseen tasks.

instruction-tuned models had no exposure to any of these seven tasks during training.

The CKA similarity results in Figure 7 reveal an interesting pattern. For the lower layers (up to layer 12), the Llama 2 model exhibited slightly higher CKA similarities than Llama 2-SFT for several tasks, similar to what we find in §4.2. This indicates that while the Llama 2-SFT model was not trained using these tasks, it produced more divergent representations in lower layers and thus more general than the ones produced by Llama 2 (we refer the reader to shared layers discussion in §4.2 for more details). However, as we move to the middle and higher layers responsible for encoding more specialized, task-specific knowledge, the Llama 2-SFT model began matching and ultimately surpassing the CKA similarities of the Llama 2 model. We can also see high variances between task similarities for both models, showing that we can not identify transition layers for Llama 2-SFT in this setup, just shared and refinement layers. These findings suggest that in addition to being trained on instructions, instruction-tuned models benefit from more general and thus better feature representations in their lower layers, which boost their performance for unseen instruction-based tasks compared to pre-trained LLMs.

5 Discussion

Our study offers comprehensive insights into the impact of instruction tuning on the representations learned by LLMs. Previous work has discussed the benefits of instruction tuning (Wei et al., 2022b; Chung et al., 2022; Longpre et al., 2023), but ours is the first to analyze their effects from a representational perspective.

Our analysis revealed that LLMs instruction-tuned on multiple tasks learned different representations in the lower layers compared to LLMs tuned on individual tasks. Similar to MTL, such representations can be shared and used across tasks (Maurer et al., 2016). Our analysis uncovered a key novel finding – we observed clear differences between pre-trained and instruction-tuned models, with the most significant representational transformations occurring in the middle transitional layers. This finding highlights the critical role of middle layers in encoding the specialized task knowledge induced by instruction tuning. Similarly, previous studies in multilingual settings have also identified language-neutral transformations in the middle layers of the network (Muller et al., 2021; Zhao et al., 2023). Furthermore, our analysis suggests that in the refinement layers, instruction-tuned models continue to shape representations toward specific tasks but without substantial representational changes with respect to task-specific information. Overall, our finding about functionality for different layers in LLMs generally aligns with previous findings on BERT, which have shown that lower layers are more general, while upper layers are known to be more task-specific (Rogers et al., 2020; Merchant et al., 2020).

Our correlation analysis also revealed insights into the relationship between representations and task complexity. Instruction-tuned models exhibited a positive correlation with reading complexity measures in the transition and refinement layers, suggesting better encoding of task-specific information for tasks with more specific vocabulary – a capability not observed in pre-trained models. Notably, instruction tuning enabled models to preserve and enhance task-specific information across a broader range of layers, as evidenced by higher

CKA similarities compared to control models. Our evaluation of unseen tasks further underscored the benefits of instruction tuning for improving generalization, with instruction-tuned models outperforming their pre-trained counterparts in deeper layers responsible for encoding complex task knowledge. This aligns with empirical evidence from Wei et al. (2022a) but also highlights how representational changes facilitated by instruction tuning strengthen cross-task transfer capabilities.

6 Conclusion

Our study used several analyses to investigate how instruction tuning shapes representations in LLMs. These analyses revealed that unlike the pre-trained LLM (Llama 2), the instruction-tuned model (Llama 2-SFT) retained a high amount of task-specific information for all tasks from the middle layers onward. Moreover, we were able to map the layers of Llama 2-SFT into three groups based on their functionality: shared layers (layers 1-9), transition layers (10-15), and refinement layers (16-32). In addition to expanding our understanding of LLMs, such mapping can greatly benefit future research in the fields of PEFT, MTL, and model compression. We also demonstrated that our mapping does not generalize to unseen tasks, revealing that a potential additional reason for the strong generalization capabilities of instruction-tuned models to unseen tasks can be related to their multi-task nature of producing more general representations.

Limitations

While our study provides valuable insights into the impact of instruction tuning on the representations learned by LLMs, there are several limitations that should be considered.

Firstly, the instruction tuning in our experiments was implemented using LoRA instead of full fine-tuning. While LoRA is computationally efficient and effective in many scenarios, it may not capture the full range of representational changes that full fine-tuning can achieve. This limitation might have influenced the depth of insights into how instruction tuning affects the model representations.

Secondly, our study exclusively used the Llama 2 model due to limited computational resources available. Although Llama 2 is a powerful and widely used LLM, relying on a single model limits the generalizability of our findings. Different models may exhibit varied representational dynamics and responses to instruction tuning. Expanding our analysis to include multiple models from different architectures would provide a more comprehensive understanding of these effects.

Additionally, we conducted our experiments on the 7B parameter version of Llama 2. While this model size is substantial, it is not the largest available. Larger models, with their greater capacity and potentially different representational capabilities, might show different patterns in response to fine-tuning. Investigating multiple model sizes would help ascertain whether the observed trends hold across different scales.

Moreover, our experiments focused solely on NLP tasks and did not explore fine-tuning on code or other specialized domains. Coding tasks often involve unique representational challenges and might reveal different insights into the impact of fine-tuning. Including such tasks in future work would broaden the scope and applicability of our findings.

Acknowledgments

This work was supported by the UKRI Centre for Doctoral Training (CDT) in Natural Language Processing through UKRI grant EP/S022481/1. We would like to thank Bonnie Webber, Ivan Titov and the anonymous reviewers for their helpful feedback. We appreciate the use of computing resources through the CSD3 cluster at the University of Cambridge and the Baskerville cluster at the University of Birmingham.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017a. 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)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2017b. Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Rich Caruana. 1997. Multitask learning. *Mach. Learn.*, 28(1):41–75.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. 2019. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3519–3529. PMLR.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. 2016. The benefit of multitask representation learning. *Journal of Machine Learning Research*, 17(81):1–32.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487, Dublin, Ireland. Association for Computational Linguistics.

Benjamin Muller, Yanai Elazar, Benoît Sagot, and Djamé Seddah. 2021. First align, then predict: Understanding the cross-lingual ability of multilingual BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2214–2231, Online. Association for Computational Linguistics.

OpenAI et al. 2024. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035.

Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 186–195, Honolulu, Hawaii. Association for Computational Linguistics.

Yifu Qiu, Zheng Zhao, Yftah Ziser, Anna Korhonen, Edoardo M. Ponti, and Shay B. Cohen. 2024. Spectral editing of activations for large language model alignment. *Preprint*, arXiv:2405.09719.

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

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. 2017. SVCCA: singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6076–6085.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

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. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. *Preprint*, arXiv:2401.00368.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022b. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zheng Zhao, Yftah Ziser, and Shay Cohen. 2022. Understanding domain learning in language models through subpopulation analysis. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 192–209, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Zheng Zhao, Yftah Ziser, Bonnie Webber, and Shay Cohen. 2023. A joint matrix factorization analysis of multilingual representations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12764–12783, Singapore. Association for Computational Linguistics.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

A Dataset Details

This appendix provides a detailed overview of the datasets used in this study. We followed Wei et al. (2022a) and organized all tasks into the following task clusters:

- **Closed-book Question Answering (QA)** requires models to answer questions about the world without direct access to the answer-containing information.

- **Commonsense Reasoning** tests the capacity for physical or scientific reasoning infused with common sense.

- **Coreference Resolution** identifies expressions referring to the same entity within a given text.

- **Natural Language Inference (NLI)** focuses on the relationship between two sentences, typically evaluating if the second sentence is true, false, or possibly true based on the first sentence.

- **Paraphrase Detection** involves evaluating if two sentences have the same meaning. While it can be considered a form of bidirectional entailment, it remains distinct from NLI in academic contexts.

- **Reading Comprehension** assesses the ability to answer questions based on a given passage containing the necessary information.

- **Reading Comprehension with Commonsense** merges the tasks of reading comprehension and commonsense reasoning.

- **Sentiment Analysis** is a traditional NLP task that determines whether a text expresses a positive or negative sentiment.

- **Struct-to-Text** involves generating natural language descriptions from structured data.

- **Translation** is the task of translating text from one language to another.

- **Summarization** involves creating concise summaries from longer texts.

- **Unseen** clusters uses the original miscellaneous task cluster from Wei et al. (2022a) which includes:

  1. Conversational question-answering;
  2. Evaluating context-sentence word meanings;
  3. Linguistic acceptability;
  4. Math questions;
  5. Question classification.

We provide tasks contained in each cluster in Table 1.

B Additional Results

B.1 Results on Model Evaluation

We provide the results on all control models and instruction-tuned Llama 2-SFT in Table 3 (for natural language understanding tasks) and Table 4 (for natural language generation tasks). To further evaluate the validness of our instruction tuning, we also benchmark our models on two popular benchmark datasets: MMLU (Hendrycks et al., 2021) and BBH (Suzgun et al., 2022). We provide results in Table 2. We can see that Llama 2-SFT outperforms Llama 2 on both of these benchmarks.

B.2 Results on Analysis

Here we provide additional results on our analysis. We provide the distribution of CKA similarities for all layers by tasks clusters in Figure 8 and 9. We also provide the t-SNE visualizations of representations in different layers of Llama 2 in Figure 10. Lastly, we provide the same visualizations for Llama 2-SFT in Figure 11.

Task Cluster	**Dataset**	**Task Cluster**	**Dataset**
Natural language inference	ANLI CB MNLI QNLI SNLI WNLI RTE	Reading comprehension	BoolQ DROP MultiRC OBQA SQuADv1 SQuADv2
Commonsense reasoning	COPA HellaSwag PiQA StoryCloze	Sentiment analysis	IMDB Sentiment140 SST-2 Yelp
Closed-book QA	ARC NQ TriviaQA	Paraphrase detection	MRPC QQP Paws Wiki STS-B
Coreference resolution	DPR Winogrande WSC273	Reading comprehension with commonsense	CosmosQA ReCoRD
Struct to text	CommonGen DART E2ENLG WebNLG	Translation	En–Fr from WMT'14 WMT'16 En–Es from Paracrawl
Summarization	AESLC CNN-DM Gigaword MultiNews Newsroom Samsum XSum AG News Opinion Abstracts - Rotten Tomatoes Opinion Abstracts - iDebate Wikilingua English	Unseen	CoQA QuAC WiC TREC CoLA Math questions

Table 1: Dataset details grouped by task clusters. For WMT'16, we include En–De, En–Tr, En–Cs, En–Fi, En–Ro, and En–Ru translation pairs. For all details about each dataset including the dataset size, please refer to Wei et al. (2022a).

	MMLU	**BBH**
Llama 2	41.25	32.82
Llama 2-SFT	47.81	37.49

Table 2: Results for Llama 2 and Llama 2-SFT on MMLU and BBH. We use a 0-shot evaluation for MMLU to assess our models. For BBH, we follow the default evaluation protocol and use a 3-shot evaluation.

Dataset	**Metric**	**Result**	
		Llama 2-SFT	Control Model
Natural Language Inference			
ANLI (r1)	Accuracy	51.87	54.45
ANLI (r2)	Accuracy	49.45	55.85
ANLI (r3)	Accuracy	47.48	54.14
CB	Accuracy	49.59	83.17
MNLI (matched)	Accuracy	87.25	88.64
MNLI (mismatched)	Accuracy	87.72	89.41
QNLI	Accuracy	83.00	86.46
SNLI	Accuracy	82.96	84.06
WNLI	Accuracy	71.22	69.64
RTE	Accuracy	81.52	81.21
Reading Comprehension			
BoolQ	Accuracy	83.53	88.18
DROP	F1	44.42	52.05
MultiRC	F1	72.19	73.92
OBQA	Accuracy	64.92	65.37
SQuADv1	F1	73.91	74.24
SQuADv2	F1	22.75	23.55
Commonsense Reasoning			
COPA	Accuracy	83.56	76.97
HellaSwag	Accuracy	71.43	73.49
PiQA	Accuracy	78.21	78.43
StoryCloze	Accuracy	85.81	84.82
Sentiment Analysis			
IMDB	Accuracy	72.06	74.54
Sentiment140	Accuracy	45.52	44.53
SST-2	Accuracy	79.14	79.03
Yelp	Accuracy	74.35	74.40
Closed-book QA			
ARC (Challenge)	Accuracy	59.09	52.83
ARC (Easy)	Accuracy	67.18	65.72
TriviaQA	Accuracy	59.00	59.26
NQ	Accuracy	28.79	31.18
Paraphrase Detection			
MRPC	Accuracy	78.35	84.73
QQP	Accuracy	84.91	87.37
PAWS Wiki	Accuracy	91.77	94.15
STS-B	Accuracy	47.46	51.20
Coreference Resolution			
DPR	Accuracy	85.12	72.53
Winogrande	Accuracy	69.68	69.93
WSC273	Accuracy	55.78	47.24
Read. Comp. w/ Commonsense			
CosmosQA	Accuracy	66.60	69.36
ReCoRD	Accuracy	85.13	85.78
Unseen			
CoQA	Accuracy	66.60	73.93
QuAC	Accuracy	18.29	33.99
WiC	Accuracy	56.47	70.77
TREC	Accuracy	57.05	80.25
CoLA	Accuracy	34.85	70.91
Math Questions	Accuracy	4.43	35.50

Table 3: Performance metrics grouped by natural language understanding task clusters for Llama 2-SFT and control models (Llama 2 model individually fine-tuned on each task). “Read. Comp. w/ Commonsense” denotes reading comprehension with commonsense.

Dataset	**Metric**	**Result**	
		Llama 2-SFT	Control Model
Struct-to-Text			
CommonGen	ROUGE-L	45.92	46.52
DART	ROUGE-L	55.46	57.28
E2ENLG	ROUGE-L	50.17	50.96
WebNLG	ROUGE-L	62.92	65.22
Translation			
WMT'14 En–Fr	BLEU	59.30	59.29
WMT'16 En–De	BLEU	56.84	57.45
WMT'16 En–Tr	BLEU	39.41	43.58
WMT'16 En–Cs	BLEU	46.92	47.21
WMT'16 En–Fi	BLEU	48.57	50.28
WMT'16 En–Ro	BLEU	56.03	57.70
WMT'16 En–Ru	BLEU	51.41	52.12
ParaCrawl En–Es	BLEU	54.76	56.39
Summarization			
AESLC	ROUGE-L	29.98	31.68
CNN-DM	ROUGE-L	17.38	19.59
Gigaword	ROUGE-L	28.69	30.22
MultiNews	ROUGE-L	15.17	16.61
Newsroom	ROUGE-L	18.95	22.43
Samsum	ROUGE-L	36.36	37.72
XSum	ROUGE-L	25.51	29.57
AG News	ROUGE-L	77.26	80.99
Opinion Abstracts - Rotten Tomatoes	ROUGE-L	19.36	21.70
Opinion Abstracts - iDebate	ROUGE-L	18.90	23.14
Wikilingua English	ROUGE-L	30.22	32.18

Table 4: Performance metrics grouped by natural language generation task clusters for Llama 2-SFT and control models (Llama 2 model individually fine-tuned on each task).

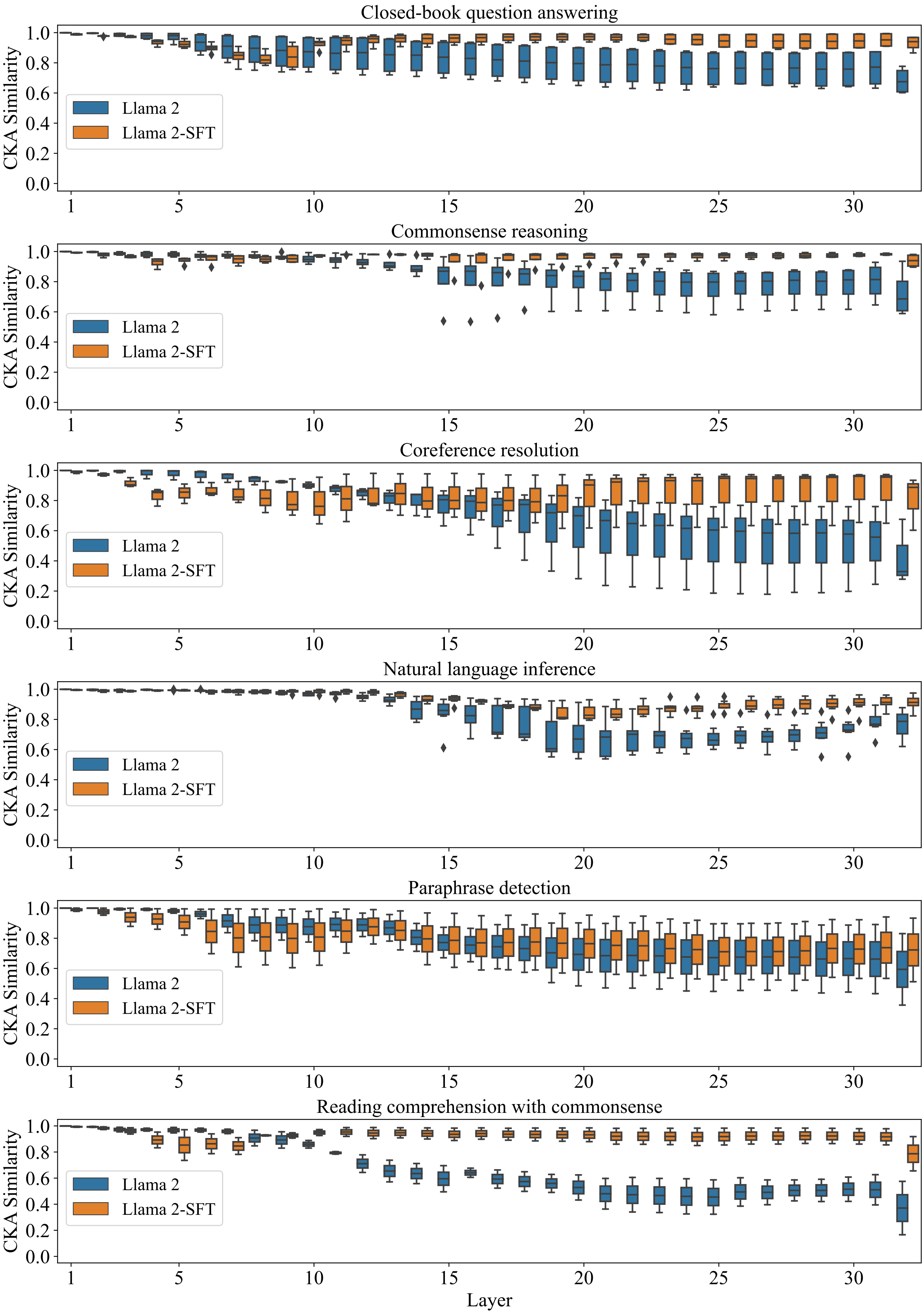


Figure 8: Distribution of CKA similarities across all layers for the pre-trained Llama 2 model and the instruction-tuned Llama 2-SFT model, grouped by different task clusters.

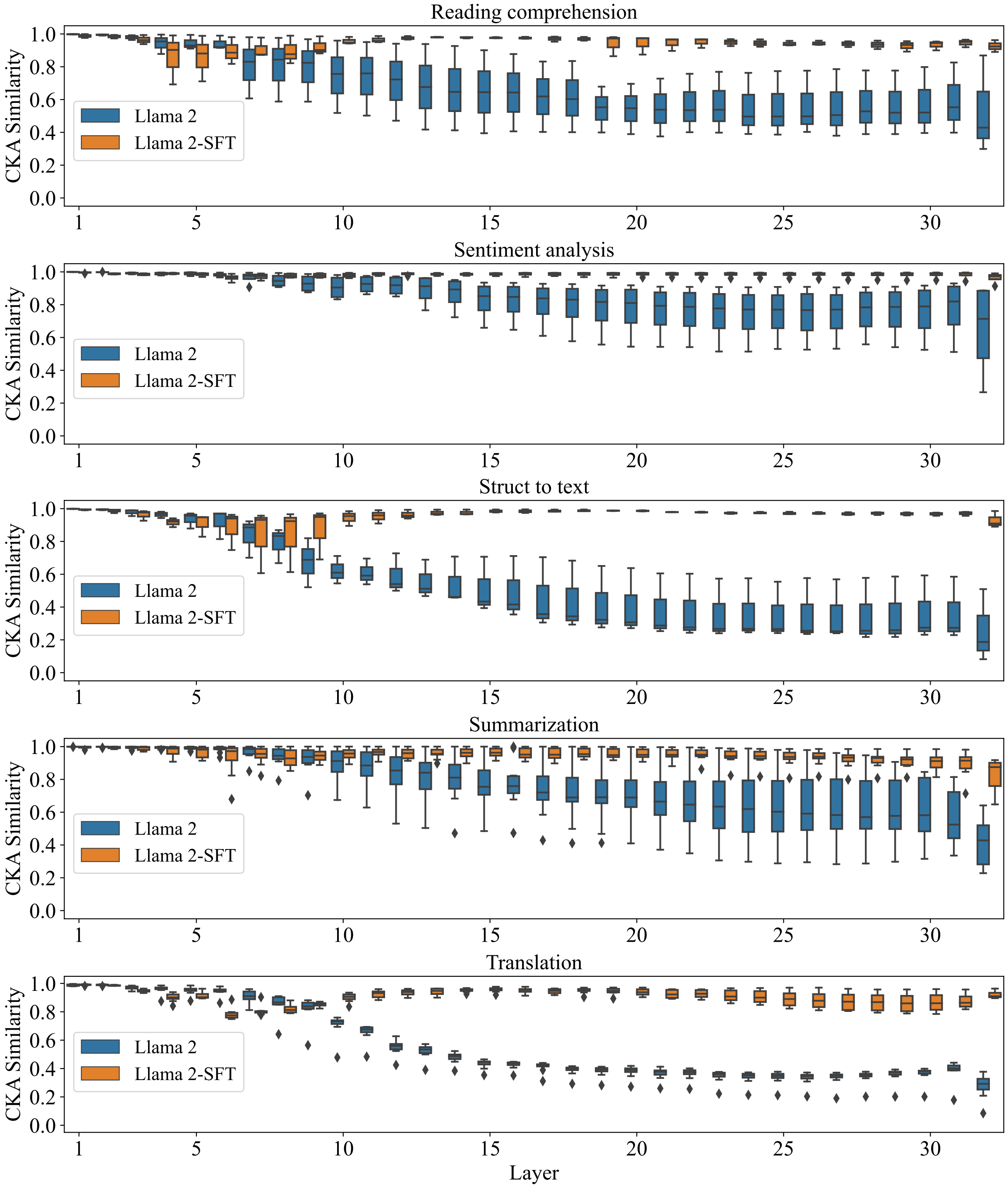
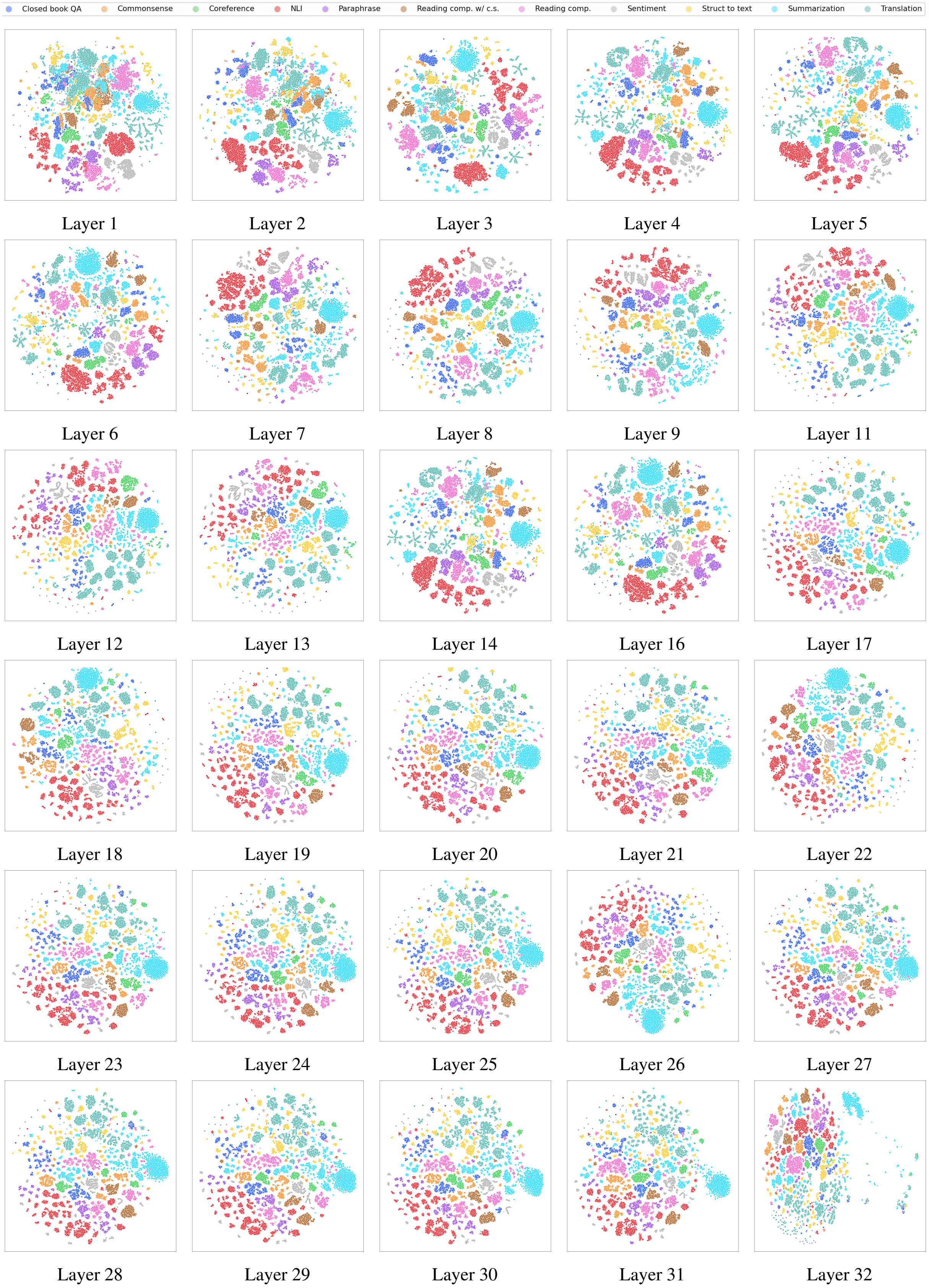


Figure 9: Distribution of CKA similarities across all layers for the pre-trained Llama 2 model and the instruction-tuned Llama 2-SFT model, grouped by different task clusters.



Layer 1 Layer 2 Layer 3 Layer 4 Layer 5

Layer 6 Layer 7 Layer 8 Layer 9 Layer 11

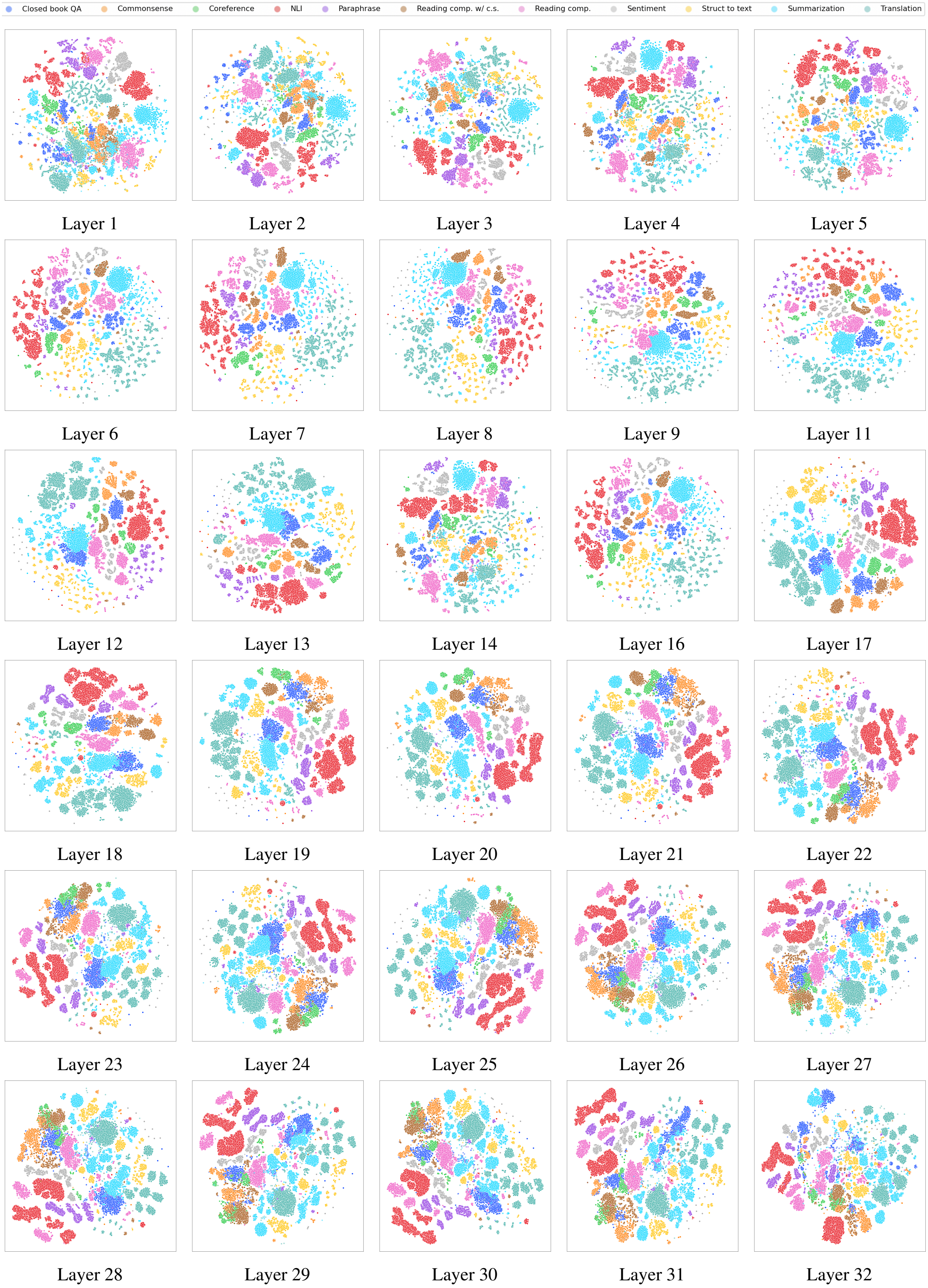
Layer 12 Layer 13 Layer 14 Layer 16 Layer 17

Layer 18 Layer 19 Layer 20 Layer 21 Layer 22

Layer 23 Layer 24 Layer 25 Layer 26 Layer 27

Layer 28 Layer 29 Layer 30 Layer 31 Layer 32

Figure 10: t-SNE visualizations of the representations for each task cluster in different layers of the pre-trained Llama 2 model. Each subplot presents the t-SNE projection of the representations, color-coded by task cluster, for a specific layer of the respective model. “Reading comp.” denotes reading comprehension tasks, and “reading comp. w/ c.s.” denotes reading comprehension tasks with commonsense reasoning. We omit layer 10 and 15 to fit in one page and as we have provided them earlier.



Layer 1 Layer 2 Layer 3 Layer 4 Layer 5

Layer 6 Layer 7 Layer 8 Layer 9 Layer 11

Layer 12 Layer 13 Layer 14 Layer 16 Layer 17

Layer 18 Layer 19 Layer 20 Layer 21 Layer 22

Layer 23 Layer 24 Layer 25 Layer 26 Layer 27

Layer 28 Layer 29 Layer 30 Layer 31 Layer 32

Figure 11: t-SNE visualizations of the representations for each task cluster in different layers of the instruction-tuned Llama 2-SFT model. Each subplot presents the t-SNE projection of the representations, color-coded by task cluster, for a specific layer of the respective model. “Reading comp.” denotes reading comprehension tasks, and “reading comp. w/ c.s.” denotes reading comprehension tasks with commonsense reasoning. We omit layer 10 and 15 to fit in one page and as we have provided them earlier.