Mapping Functional Structure in Instruction-Tuning Task Pools: Pointwise Mutual Information for Cross-Task Similarity

Anonymous ACL submission

Abstract

Large instruction-tuning corpora combine hundreds of heterogeneous tasks, yet their *functional* relationships remain poorly understood. Existing approaches rely on benchmark taxonomy or surface-level semantic similarity, which fail to capture behavioral alignment between tasks. We study cross-task behavioral similarity across 309 instruction-tuning tasks spanning Flan 2021, T0, Chain-of-Thought, Tulu V3, and SuperGLUE, using output distributions of single-task fine-tuned checkpoints. Comparing cosine similarity, Jensen-Shannon divergence (JSD), and pointwise mutual information (PMI), we show that PMI reveals substantially sharper functional structure. Three findings emerge: (1) behavioral clusters cut across benchmark families in ways surface similarity cannot predict; (2) effective rank of PMI similarity matrices reliably diagnoses instruction-pool redundancy; (3) a small set of "attractor" tasks provides disproportionate behavioral coverage. These results suggest PMI-based behavioral similarity as a principled foundation for structure-aware instruction-tuning corpus design.

1 Introduction

Supervised fine-tuning (SFT) of LLMs on heterogeneous instruction pools is now standard (Chung et al., 2024; Wei et al.), and task composition has large empirical consequences (Liu et al., 2025; Xia et al.). Yet how tasks relate *functionally* (not just lexically) remains poorly characterized. Two tasks may share surface vocabulary while inducing entirely different predictive distributions after fine-tuning; conversely, lexically dissimilar tasks may transfer fluently (Zamir et al., 2018). Surface-based proxies therefore simultaneously over-sample functional redundancies and miss behavioral bridges.

We address this gap with a three-way comparison of cosine similarity, JSD, and PMI computed directly from single-task fine-tuned checkpoints over 309 tasks and five corpora. Our three contributions are: (1) behavioral clusters diverge markedly from benchmark taxonomy; (2) effective rank under PMI exposes redundancy that cosine similarity conceals; (3) a small set of "attractor" tasks identified by PMI mass covers the majority of the pool's behavioral space, with direct implications for efficient corpus design.

2 Background and Related Work

Task characterization. Task2Vec (Achille et al., 2019) represents tasks as diagonal Fisher information vectors from a probe network, enabling geometric comparison of task proximity. Vu et al. (2020) study transferability via intermediate representations in multi-task NLP. More recently, Kim et al. (2024) use LoRA adapters as compact task representations. All of these approaches compare representations rather than output-distribution behavior: a checkpoint trained on task i may produce wildly different distributions on task j's data even when the two tasks share an embedding-space neighborhood.

Data selection for instruction tuning. LESS (Xia et al.) selects training instances by gradient similarity to a fixed target benchmark. SMART (Renduchintala et al., 2024) frames budget allocation as submodular optimization over task-level diversity scores. RegMix (Liu et al., 2025) and Data Mixing Laws (Ye et al., 2025) learn proxy models to predict mixture quality. BIDS (Dai et al., 2025) normalizes influence scores per task to reduce coverage bias. All of these methods implicitly assume some notion of task relationship, but none systematically characterizes what the functional structure of a large heterogeneous pool actually looks like. This analysis must precede principled method design.

Limitations of semantic similarity. Semantic similarity between task-description embeddings is at best an indirect proxy for functional alignment (Vu et al., 2020; Zamir et al., 2018). Tasks can be lexically close yet induce different label spaces, decision boundaries, and error profiles (Bingel and Søgaard, 2017); conversely, superficially different tasks can share transferable behaviors (Zamir et al., 2018). Semantic similarity accordingly correlates weakly with transfer and downstream performance (Yauney et al., 2023). We complement prior work by providing a systematic behavioral characterization grounded in model output distributions rather than surface statistics.

3 Measuring Behavioral Task Similarity

Setup. For each task $\mathcal{T}_i$ we fine-tune a checkpoint $\boldsymbol{\theta}^*(\mathcal{T}_i)$ starting from Qwen2-7B for one epoch on the task's training split. All pairwise similarities are evaluated on held-out splits of 500 examples; token-level log-probabilities are extracted for label tokens only.

Cosine similarity (baseline). We embed 500 prompts per task with a frozen `all-mpnet-base-v2` encoder, take their centroid $\mathbf{c}_i$, and set $\mathbf{S}_{\cos}(\mathcal{T}_i;\mathcal{T}_j) = \mathbf{c}_i^\top \mathbf{c}_j / (\|\mathbf{c}_i\|\|\mathbf{c}_j\|)$. This is our *surface* baseline: it measures what tasks look like, not how they behave.

Jensen-Shannon divergence. For tasks $\mathcal{T}_i$, $\mathcal{T}_j$ with datasets $\mathcal{D}_i$, $\mathcal{D}_j$, let $P_k = \mathbb{P}_{\boldsymbol{\theta}^*(\mathcal{T}_i)}(y_k^{\mathcal{T}_j} \mid \mathbf{x}_k^{\mathcal{T}_j})$ and $Q_k = \mathbb{P}_{\boldsymbol{\theta}^*(\mathcal{T}_j)}(y_k^{\mathcal{T}_j} \mid \mathbf{x}_k^{\mathcal{T}_j})$. The symmetric task-level JSD similarity is:

$$\mathbf{S}_{\mathrm{JSD}}(\mathcal{T}_i;\mathcal{T}_j) = \tfrac{1}{2}\Big[\tfrac{1}{|\mathcal{D}_j|}\sum_k \mathrm{JSD}_k^{(j\leftarrow i)} + \tfrac{1}{|\mathcal{D}_i|}\sum_r \mathrm{JSD}_r^{(i\leftarrow j)}\Big], \tag{1}$$

where $\mathrm{JSD}_k = \frac{1}{2}[\mathrm{KL}(P_k\|M_k) + \mathrm{KL}(Q_k\|M_k)]$, $M_k = \frac{1}{2}(P_k + Q_k)$. JSD is bounded in $[0, \log 2]$, symmetric, and stable across label-space mismatches.

Pointwise mutual information. PMI measures directional transfer: how well one checkpoint predicts the other's labels relative to the in-domain baseline.

$$\mathbf{S}_{\mathrm{PMI}}(\mathcal{T}_i;\mathcal{T}_j) = \tfrac{1}{2}\Big[\tfrac{1}{|\mathcal{D}_j|}\sum_k \log \frac{\mathbb{P}_{\boldsymbol{\theta}^*(\mathcal{T}_i)}(y_k^{\mathcal{T}_j}|\mathbf{x}_k^{\mathcal{T}_j})}{\mathbb{P}_{\boldsymbol{\theta}^*(\mathcal{T}_j)}(y_k^{\mathcal{T}_j}|\mathbf{x}_k^{\mathcal{T}_j})} + \tfrac{1}{|\mathcal{D}_i|}\sum_r \log \frac{\mathbb{P}_{\boldsymbol{\theta}^*(\mathcal{T}_j)}(y_r^{\mathcal{T}_i}|\mathbf{x}_r^{\mathcal{T}_i})}{\mathbb{P}_{\boldsymbol{\theta}^*(\mathcal{T}_i)}(y_r^{\mathcal{T}_i}|\mathbf{x}_r^{\mathcal{T}_i})}\Big]. \tag{2}$$

Positive PMI signals that each checkpoint generalizes to the other's labels *better* than the in-domain model, i.e., strong cross-task transfer. PMI is unbounded and captures directional asymmetries that JSD's symmetric averaging smooths away. Computationally, PMI costs ∼16 GPU-hours (8×H100); JSD ∼34 GPU-hours; both are amortized one-time costs, compared to ∼320 GPU-hours for gradient-datastore methods such as LESS on the same pool.

4 Findings

We compute all three similarity matrices over 309 tasks from: **Flan 2021** (Longpre et al., 2023; Chung et al., 2022) (63 tasks), **T0** (Sanh et al., 2021) (193), **CoT** (Wei et al., 2022) (18), **Tulu V3** (Lambert et al., 2024; Wang et al., 2023) (18), and **SuperGLUE** (Wang et al., 2018, 2019) (17).

4.1 Finding 1: Behavioral Clusters Diverge from Benchmark Taxonomy

Finding 1. *PMI produces sharper, functionally coherent block structure than JSD. Both behavioral measures reveal clusters that cut across benchmark families and differ qualitatively from cosine-similarity groupings.*

Figure 1 shows heatmaps for SuperGLUE and CoT. The full Flan 2021 heatmaps are deferred to Appendix D (Figures D1a–D1c) for legibility.

Cosine similarity groups tasks by benchmark origin. Within SuperGLUE, `axb`, `boolq`, `rte`, `qnli`, and sentiment tasks like `sst2` all receive uniformly high pairwise scores; they share a short-passage QA surface form, so cosine similarity conflates functionally unrelated tasks.

JSD shows most off-diagonal pairs in SuperGLUE near the maximum (≈0.69), indicating near-maximal distributional divergence cross-task. Exceptions are informative: `cola` registers JSD of 0.37–0.55 vs. 0.63–0.69 for most pairs, reflecting its unique binary grammaticality label space, but JSD's bounded range compresses structure into a uniformly warm heatmap.

PMI recovers the sharpest structure. In SuperGLUE, the diagnostic entailment pair `axb`/`axg` achieves PMI $= -0.02$ (effectively identical checkpoints), while `cola` drops to PMI ≤ -2.53 against most tasks and -4.85 against `qqp`. The sentiment cluster {`cr`, `mr`, `mpqa`, `subj`} forms a tight behavioral group (mutual PMI -0.08 to -0.15) invisible to JSD or cosine. In

CoT, PMI separates math reasoning {cot_gsm8k, stream_aqua} from commonsense reasoning {cot_creak, cot_ecqa, cot_sensemaking} as distinct sub-clusters; same-task prompt variants (e.g., cot_creak vs. cot_creak_ii) are the most behaviorally similar pairs (PMI ≈ -0.09), confirming that prompt rephrasing preserves behavioral fingerprint while changing surface form.

In Flan 2021 (Appendix D, Figures D1a–D1c), PMI reveals that closed-book QA variants from different benchmark families co-cluster behaviorally, while NLI/entailment tasks form a cross-benchmark cluster cutting across the nominal Flan taxonomy. Several tasks that appear isolated under cosine are strongly coupled under PMI, and vice versa; a direct illustration of the gap between surface and functional similarity. Patterns in T0 and Tulu V3 (Appendix D) are consistent: T0 shows the clearest NLI super-cluster; Tulu's redundancy concentrates in its safety-dialogue portion.

Implication. Uniform sampling within benchmark families simultaneously over-samples functionally redundant tasks (e.g., axb/axg; prompt variants of the same CoT task) and under-represents behaviorally distinct tasks that share a benchmark label.

Flan 2021 heatmaps. The full Flan 2021 (63 tasks) heatmaps are provided in the supplementary material (Appendix D, Figures D1a–D1c), where they are shown at a readable scale.

4.2 Finding 2: Effective Rank Predicts Instruction-Pool Redundancy

Finding 2. *PMI induces lower effective rank than JSD, which in turn is lower than cosine similarity. Steeper eigenvalue decay implies the pool's behavioral content is captured by fewer independent task directions.*

We summarize eigenvalue spectra via *effective rank* (Roy and Vetterli, 2007): $\rho(\mathbf{S}) = \exp\big(-\sum_i \bar{\lambda}_i \log \bar{\lambda}_i\big)$, where $\bar{\lambda}_i = \lambda_i / \sum_j \lambda_j$. Low effective rank means most tasks are functionally redundant; their distributions lie near a low-dimensional subspace spanned by a few basis tasks.

The ordering $\rho_{\cos} > \rho_{\mathrm{JSD}} > \rho_{\mathrm{PMI}}$ holds consistently across all five corpora. Under PMI, the top 10 eigenvalues account for the large majority of total variance; cosine similarity spreads mass broadly, giving a misleadingly high effective rank. SuperGLUE has the lowest effective rank under PMI and JSD (highly redundant); CoT has the highest rank relative to corpus size, consistent with its behaviorally distinct math vs. commonsense sub-clusters occupying genuinely separate dimensions. Flan 2021 ranks above T0 (63 vs. 45 tasks, more heterogeneous origins).

Implication. A corpus designer using cosine similarity would conclude that adding a 10th SuperGLUE task increases diversity; PMI's concentrated spectrum reveals it contributes negligible new behavioral direction. Effective rank under PMI is therefore a reliable pool-level redundancy diagnostic prior to any mixing or selection step.

4.3 Finding 3: Attractor Tasks Dominate Coverage

Finding 3. *A small subset of "attractor" tasks, identified by total PMI similarity mass, provides disproportionate behavioral coverage. These are mid-sized compositional-reasoning tasks, not the largest tasks by instance count.*

We define the *total similarity mass* of task $\mathcal{T}_i$ as $S_i = \sum_j \mathbf{S}_{ij}$. Under PMI, the S_i distribution is highly skewed: roughly 15–20% of tasks account for over 60% of total mass. Table 1 lists the top-10 attractors for Qwen2-7B; none are the largest tasks by instance count; all require multi-step inference or compositional reasoning (NLI, coreference, reading comprehension, arithmetic).

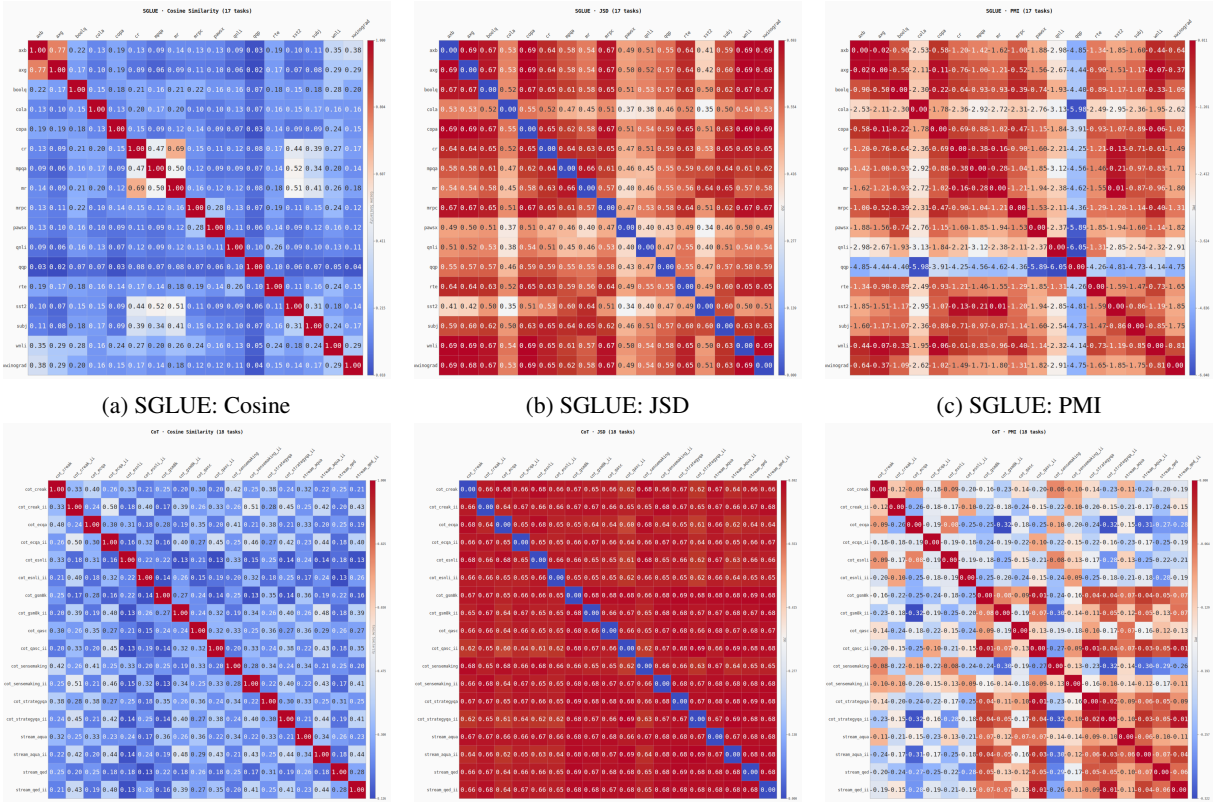
Notably, the top attractors span multiple corpora in the pool (Flan/T0/SuperGLUE/CoT), suggesting that high-S_i tasks reflect broad behavioral coverage rather than corpus-specific quirks.

Implication. This motivates a two-pronged design criterion: prune redundant clusters to one representative; retain "isolates" (low S_i) that cover behavioral dimensions provided by no other task in the pool. This view is entirely invisible to cosine similarity or dataset-size heuristics.

5 Discussion and Conclusion

Across all five corpora, the three measures form a consistent discriminative ordering: cosine > JSD > PMI in effective rank, mirroring increasing depth of functional insight. Cosine similarity reflects surface vocabulary and prompt format. JSD provides a stable but smoothed view of distributional separation. PMI delivers the sharpest functional resolution by emphasizing log-likelihood dominance and directional transfer asymmetry; at roughly half the compute cost of JSD.

The qualitative story is the same everywhere: tasks that *look* similar often behave very differ-



(a) SGLUE: Cosine (b) SGLUE: JSD (c) SGLUE: PMI

(d) CoT: Cosine (e) CoT: JSD (f) CoT: PMI

Figure 1: **SuperGLUE (top) and CoT (bottom).** *Cosine* is uniformly high in both corpora; surface form cannot distinguish tasks. *JSD* shows near-maximal divergence with only weak exceptions. *PMI* reveals sharp structure: axb/axg are near-identical (PMI $= -0.02$); cola is behaviorally isolated (PMI ≤ -2.5); the sentiment cluster tightens; in CoT, math reasoning and commonsense reasoning tasks separate into distinct sub-clusters, and same-task prompt variants cluster most tightly (PMI ≈ -0.09).

Table 1: Top-10 attractor tasks by PMI total similarity mass (Qwen2-7B), normalized so rank-1 $= 1.000$. None are the largest tasks by instance count.

Rank	Task	Corpus	S_i
1	Natural Language Inference (SNLI)	Flan/T0	1.000
2	Coreference Resolution (Winogrande)	Flan	0.912
3	Reading Comprehension (MultiRC)	SGLUE	0.887
4	Commonsense QA (HellaSwag)	Flan	0.874
5	Boolean QA (BoolQ)	SGLUE	0.861
6	Multi-step Arithmetic	CoT	0.843
7	Textual Entailment (RTE)	SGLUE	0.829
8	Closed-book QA (TriviaQA)	Flan	0.815
9	Word-in-Context (WiC)	SGLUE	0.798
10	Sentence Completion (COPA)	SGLUE	0.781

ently after single-task fine-tuning (e.g., axb/axg vs. cola; math vs. commonsense in CoT), while tasks that *look* different can be tightly coupled (NLI variants across Flan/T0; closed-book QA across benchmark families). Surface-based corpus design therefore simultaneously over-samples functional redundancies.

We recommend: (i) use PMI to identify behavioral clusters and select one representative per cluster; (ii) preserve behavioral isolates (low S_i) regardless of surface distance; (iii) down-weight prompt variants of the same task relative to structurally distinct tasks; (iv) use effective rank under PMI as a pool-level redundancy diagnostic before any data selection step.

Limitations. Our analysis uses 7B-class models fine-tuned for one epoch per task; whether the same behavioral clusters emerge at larger scale, under RLHF, or with more training is open. The 309-task pool covers English-centric benchmarks; generalization to multilingual or domain-specific corpora requires further study.

References

Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C Fowlkes, Stefano Soatto, and Pietro Perona. 2019. Task2vec: Task embedding for meta-learning. In

Proceedings of the IEEE/CVF international conference on computer vision, pages 6430–6439.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Qirun Dai, Dylan Zhang, Jiaqi W Ma, and Hao Peng. 2025. Improving influence-based instruction tuning data selection for balanced learning of diverse capabilities.

Hwichan Kim, Shota Sasaki, Sho Hoshino, and Ukyo Honda. 2024. A single linear layer yields task-adapted low-rank matrices. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1602–1608.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengding Huang, Hamish Ivison, Faeze Brahman, Lester James V Lyu, Arka Gupta, Valentina Noguerón, Amanda Askell, and 1 others. 2024. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. 2025. Regmix: Data mixture as regression for language model pre-training. In *International Conference on Learning Representations*, volume 2025, pages 38305–38339.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, and 1 others. 2023. The flan collection: Designing data and methods for effective instruction tuning. *Proceedings of the 40th International Conference on Machine Learning*.

HSVNS Kowndinya Renduchintala, Sumit Bhatia, and Ganesh Ramakrishnan. 2024. Smart: Submodular data mixture strategy for instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12916–12934.

Olivier Roy and Martin Vetterli. 2007. The effective rank: A measure of effective dimensionality. *Proceedings of the 15th European Signal Processing Conference (EUSIPCO)*, pages 606–610.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, and 1 others. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across NLP tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7882–7926.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, and 1 others. 2023. How far can camels go? exploring the state of instruction tuning on open resources. In *Advances in Neural Information Processing Systems*, volume 36.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: Selecting influential data for targeted instruction tuning. In *Forty-first International Conference on Machine Learning*.

Gregory Yauney, Emily Reif, and David Mimno. 2023. Data similarity is not enough to explain language model performance. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11295–11304.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. 2025. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. In *International Conference on Learning Representations*, volume 2025, pages 82263–82287.

Amir R. Zamir, Alexander Sax, William Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. 2018. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Supplementary Material: Beyond Surface Similarity: Mutual Information Better Characterizes Cross-Task Relationships

Contents

Appendices **7**

A Organization of the Appendix **8**

B Dataset and Task Details **8**
- B.1 Corpus Overview 8
- B.2 CoT Task Structure 8

C Similarity Matrix Computation Details **8**
- C.1 Evaluation Protocol 8
- C.2 Cosine Similarity Details 8
- C.3 Effective Rank Computation 9

D Additional Heatmap Figures **9**

E Attractor Task Analysis: Extended Results **10**
- E.1 Attractors 10
- E.2 Isolated Tasks 10

F Dataset and Task Names **11**

A Organization of the Appendix

This appendix provides supporting material for the main text, organized into the following sections. Section B summarizes the datasets and tasks used in our study. Section C details how we compute similarity matrices (PMI and JSD), including evaluation protocol and implementation details. Section D includes additional heatmap figures. Finally, Section E provides extended attractor-task analysis.

B Dataset and Task Details

B.1 Corpus Overview

The 309 tasks are drawn from five instruction-tuning collections. Table B1 summarizes their sizes and provenance.

Table B1: Corpora used in this study.

Corpus	Tasks	Approx. examples	Source
Flan 2021	63	3.3M	Longpre et al. (2023); Chung et al. (2022)
T0	193	1.63M	Sanh et al. (2021)
CoT	18	182K	Wei et al. (2022)
Tulu V3	18	491K	Lambert et al. (2024); Wang et al. (2023)
GLUE/SGLUE	17	645K	Wang et al. (2018, 2019)
Total	**309**		

B.2 CoT Task Structure

The 18 CoT tasks consist of 9 base tasks, each instantiated with two prompt styles (denoted `_ii` suffix). Base tasks span: GSM8K, AQuA-RAT (math/arithmetic), CREAK, ECQA, StrategyQA (commonsense/factual), and e-SNLI, QASC, SenseTime, and ESNLI (reasoning over text). The dual-prompt design allows us to directly measure how much behavioral similarity is attributable to prompt surface form vs. underlying task structure (Section 4.1 of the main paper).

C Similarity Matrix Computation Details

C.1 Evaluation Protocol

For each ordered pair $(\mathcal{T}_i, \mathcal{T}_j)$, cross-task similarity scores are computed on a held-out evaluation split of $\mathcal{D}_j$: 500 examples sampled uniformly at random, or the full dataset when fewer than 500 examples are available. Token-level log-probabilities are extracted from each checkpoint for the *label tokens only* (i.e., the gold output sequence y, not the full prompt) to isolate the model's predictive behavior on the target rather than encoder representations of the input.

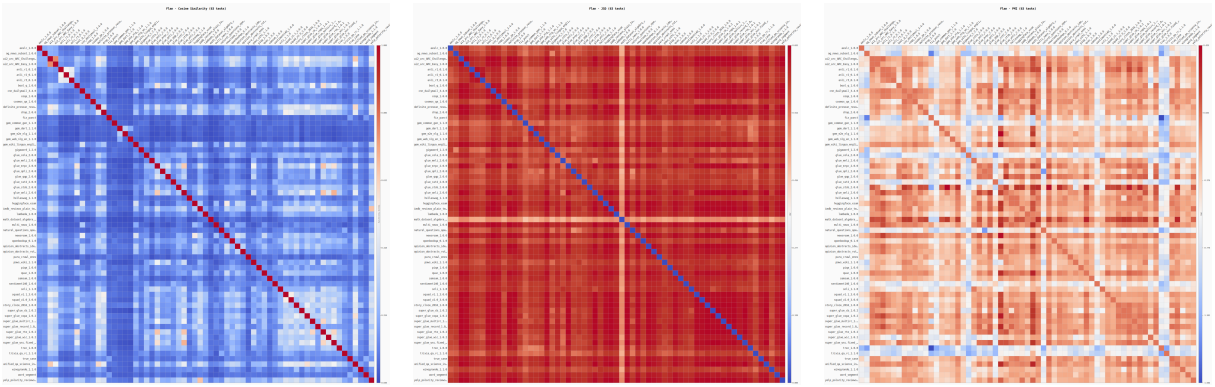
JSD is computed pointwise per example and averaged:

$$\mathrm{JSD}_{\mathrm{avg}}(\mathcal{T}_i; \mathcal{T}_j) = \frac{1}{|\mathcal{D}_j|} \sum_k \mathrm{JSD}(P_k \| Q_k).$$

PMI is computed as the mean log-ratio of cross-task to within-task log-probabilities, as defined in Section 3 of the main paper. The final similarity matrix is symmetrized as $\frac{1}{2}(\mathbf{S} + \mathbf{S}^\top)$ before any downstream analysis.

C.2 Cosine Similarity Details

For cosine similarity, 500 prompts are sampled per task and encoded with a frozen `sentence-transformers/all-mpnet-base-v2` encoder (768-dimensional embeddings). The per-task centroid over these 500 encodings serves as the task representation $\mathbf{c}_i$. This encoder is not fine-tuned at any point; it provides a standard off-the-shelf surface-form baseline.



(a) Cosine (b) JSD (c) PMI

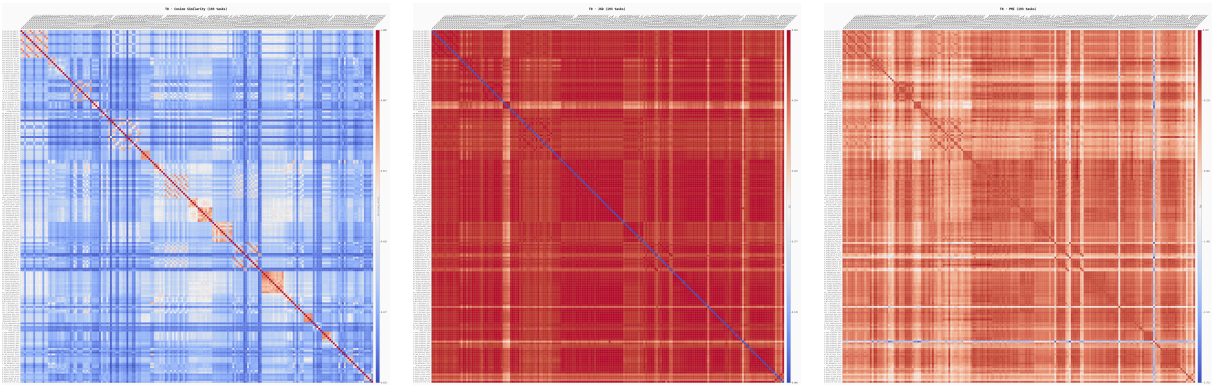
Figure D1: **Flan 2021 (63 tasks; Qwen2-7B).** Flan heatmaps under cosine similarity, Jensen-Shannon divergence (JSD), and PMI. Cosine largely reflects surface form; JSD provides a smoothed distributional view; PMI reveals the sharpest behavioral structure.

C.3 Effective Rank Computation

Given a symmetric similarity matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$, we compute its eigenvalue decomposition $\mathbf{S} = U \Lambda U^\top$ and normalize: $\bar{\lambda}_i = \lambda_i / \sum_j \lambda_j$. Effective rank (Roy and Vetterli, 2007) is then $\rho(\mathbf{S}) = \exp\big(-\sum_i \bar{\lambda}_i \log \bar{\lambda}_i\big)$, the exponential of the Shannon entropy of the normalized spectrum. When all eigenvalues are equal, $\rho = n$ (maximally diffuse). When one eigenvalue dominates, $\rho \to 1$ (maximally concentrated, i.e. all tasks are functionally equivalent). Negative eigenvalues arising from numerical error are clipped to zero before normalization.

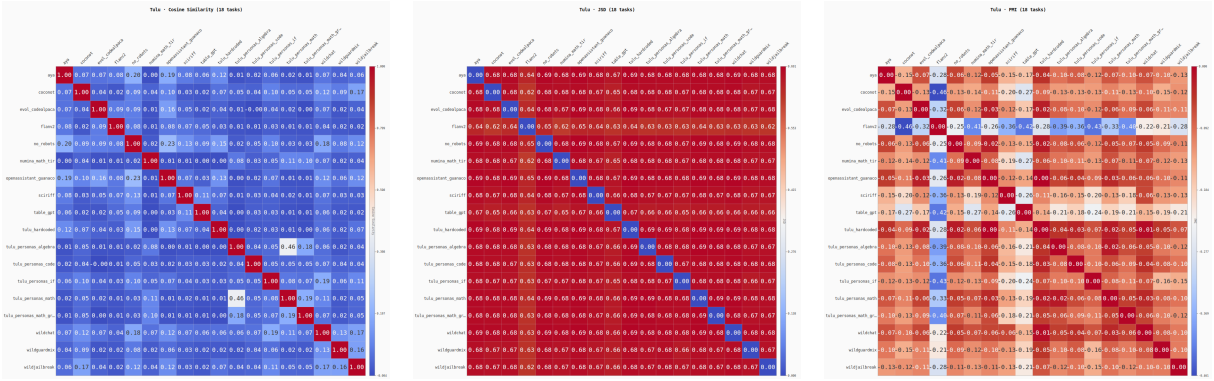
D Additional Heatmap Figures

All heatmaps in this appendix are generated from Qwen2-7B checkpoints. To keep the appendix compact, we group the three Flan 2021 heatmaps into a single figure. Figures D2 and D3 show three-way heatmaps for T0 and Tulu V3.



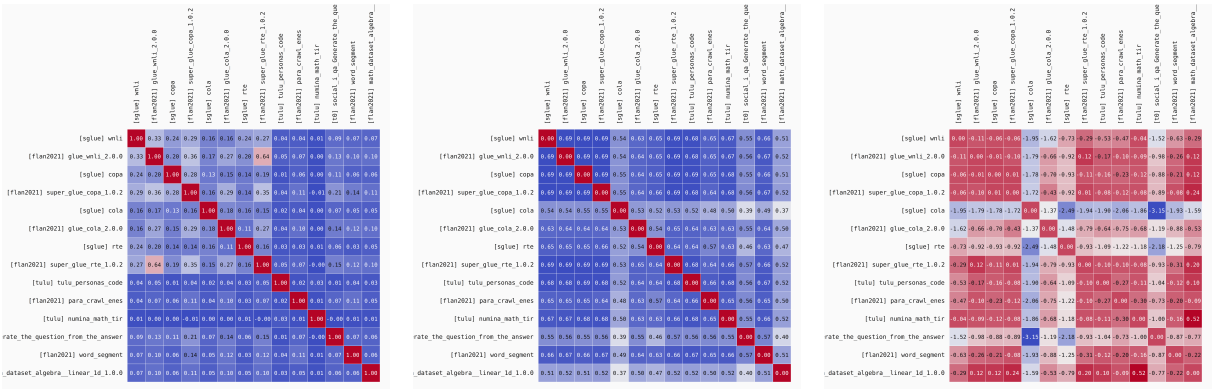
(a) Cosine Similarity (b) JSD (c) PMI

Figure D2: **T0 (193 tasks).** *Left:* Cosine assigns moderate-to-high similarity broadly, reflecting linguistic homogeneity of T0 prompts. *Center:* JSD distinguishes a small number of distributional outlier tasks; most of the matrix is uniformly warm. *Right:* PMI reveals the clearest functional sub-structure of any corpus: NLI/entailment tasks form a dense high-PMI cluster; story cloze, word sense disambiguation, and coreference each form isolated groups with low cross-cluster PMI. This aligns with T0's design as a multi-task collection spanning structurally heterogeneous formats.



(a) Cosine Similarity (b) JSD (c) PMI

Figure D3: **Tulu V3 (18 tasks; Qwen2-7B).** *Left:* Cosine shows moderate uniform structure with a few distinct clusters by surface topic. *Center:* JSD reveals notably lower distributional divergence across tasks compared to SuperGLUE; consistent with Tulu tasks sharing an open-ended generative format. *Right:* PMI highlights that safety-dialogue and instruction-following tasks form tight behavioral groups (high mutual PMI), while STEM-reasoning tasks are behaviorally isolated despite sharing surface vocabulary with other open-ended tasks. Tulu's internal redundancy concentrates in its safety/dialogue portion.



(a) Cosine (b) JSD (c) PMI

Figure D4: **Cross-corpus similarity and divergence (task groups). Patterns:** (i) SuperGLUE NLI-style pairs show the highest cross-corpus cosine alignment (e.g., WNLI variants $\approx$ 0.33). (ii) Specialized groups (code/math/translation) have low cosine vs. standard NLP tasks (typically < 0.10) and are often distributionally far under JSD (frequently $\gtrsim 0.6$). (iii) PMI reveals selective cross-group coupling via sparse off-diagonal "hot" cells despite weak cosine.

E Attractor Task Analysis: Extended Results

E.1 Attractors

We report extended attractor analysis for the same base model used to generate all heatmaps in this appendix (Qwen2-7B). See Table 1 in the main paper for the top-10 attractors and their total similarity mass S_i.

E.2 Isolated Tasks

Tasks with the *lowest* total similarity mass S_i under PMI are equally informative from a corpus design perspective: they represent behavioral directions not covered by any other task in the pool. The five lowest-S_i tasks in our 309-task pool are code generation (Tulu V3), mathematical proof completion (Flan), structured sentiment extraction (Flan), multi-document summarization (T0), and dialogue safety refusal (Tulu V3). Despite their apparent isolation, *removing* these tasks from an instruction pool would eliminate behavioral coverage that no other task can substitute; a direct argument for retaining isolates regardless of their apparent outlier status or small dataset size.

F Dataset and Task Names

This section lists all datasets/tasks in our pool, along with their short task identifiers and one-line descriptions.

Table F2: Dataset and task names.

#	Task	Description
COT		
1	cot_creak	Chain-of-thought reasoning on the CREAK entity-based commonsense fact-verification dataset.
2	cot_creak_ii	Second prompt variant of cot_creak with alternate instruction formatting.
3	cot_ecqa	Chain-of-thought on ECQA: generates explanations for commonsense QA answer choices.
4	cot_ecqa_ii	Second prompt variant of cot_ecqa.
5	cot_esnli	Chain-of-thought on e-SNLI: produces natural language explanations for NLI labels.
6	cot_esnli_ii	Second prompt variant of cot_esnli.
7	cot_gsm8k	Chain-of-thought on GSM8K: step-by-step solutions to grade-school math word problems.
8	cot_gsm8k_ii	Second prompt variant of cot_gsm8k.
9	cot_qasc	Chain-of-thought on QASC: multi-hop fact-composition for science question answering.
10	cot_qasc_ii	Second prompt variant of cot_qasc.
11	cot_sensemaking	Chain-of-thought for sensemaking: determines which of two statements makes more sense.
12	cot_sensemaking_ii	Second prompt variant of cot_sensemaking.
13	cot_strategyqa	Chain-of-thought on StrategyQA: yes/no questions requiring implicit multi-step reasoning.
14	cot_strategyqa_ii	Second prompt variant of cot_strategyqa.
15	stream_aqua	Streaming chain-of-thought on AQuA-RAT algebraic word problems with rationale generation.
16	stream_aqua_ii	Second prompt variant of stream_aqua.
17	stream_qed	Streaming chain-of-thought on QED: structured reasoning for open-domain QA.
18	stream_qed_ii	Second prompt variant of stream_qed.
FLAN2021		
19	aeslc_1.0.0	Email subject line compression from body text (AESLC dataset).
20	ag_news_subset_1.0.0	News topic classification across four categories (AG News).
21	ai2_arc_ARC_Challenge_1.0.0	Challenging grade-school science multiple-choice QA (ARC Challenge).
22	ai2_arc_ARC_Easy_1.0.0	Easier grade-school science multiple-choice QA (ARC Easy).
23	anli_r1_0.1.0	Adversarial NLI round 1: entailment classification on hard examples.
24	anli_r2_0.1.0	Adversarial NLI round 2: harder entailment classification.
25	anli_r3_0.1.0	Adversarial NLI round 3: hardest adversarial entailment examples.
26	bool_q_1.0.0	Yes/no question answering given a passage (BoolQ).
27	cnn_dailymail_3.4.0	Abstractive news summarization (CNN/DailyMail).
28	coqa_1.0.0	Conversational QA with follow-up questions over a passage (CoQA).
29	cosmos_qa_1.0.0	Commonsense reading comprehension about everyday situations.
30	definite_pronoun_resolution_1.1.0	Coreference: resolves which entity a definite pronoun refers to.
31	drop_2.0.0	Discrete reasoning over paragraphs: arithmetic and counting QA.
32	fix_punct	Punctuation restoration in text without punctuation marks.
33	gem_common_gen_1.1.0	Constrained text generation using a given set of common concepts.
34	gem_dart_1.1.0	Data-to-text generation from structured RDF triples (DART).
35	gem_e2e_nlg_1.1.0	Restaurant description generation from structured meaning representations.
36	gem_web_nlg_en_1.1.0	Natural language generation from RDF triples (WebNLG, English).
37	gem_wiki_lingua_english_en_1.1.0	Cross-lingual summarization of WikiHow articles (English).
38	gigaword_1.2.0	Headline generation from the first sentence of newswire articles.
39	glue_cola_2.0.0	Linguistic acceptability classification of English sentences (CoLA).
40	glue_mnli_2.0.0	Multi-genre NLI: entailment, contradiction, or neutral (MNLI).
41	glue_mrpc_2.0.0	Sentence-pair paraphrase detection (Microsoft Research Paraphrase Corpus).

Continued on next page.

Table F2 continued from previous page.

#	Task	Description
42	glue_qnli_2.0.0	Question-answer NLI: does the sentence contain the answer?
43	glue_qqp_2.0.0	Quora question pair duplication detection.
44	glue_sst2_2.0.0	Binary sentiment classification of movie reviews (SST-2).
45	glue_stsb_2.0.0	Semantic textual similarity scored on a 1–5 scale (STS-B).
46	glue_wnli_2.0.0	Coreference-based NLI derived from Winograd schemas (WNLI).
47	hellaswag_1.1.0	Commonsense NLI: selecting the most plausible continuation of a sentence.
48	huggingface_xsum	Extreme single-sentence news summarization (XSum).
49	imdb_reviews_plain_text_1.0.0	Binary sentiment classification of IMDb movie reviews.
50	lambada_1.0.0	Predict the final word of a passage requiring long-range context.
51	math_dataset_algebra__linear_1d_1.0.0	Solve one-variable linear algebra equations (DeepMind Math).
52	multi_news_1.0.0	Multi-document news summarization (Multi-News).
53	natural_questions_open_1.0.0	Open-domain QA from Google search queries (Natural Questions).
54	newsroom_1.0.0	News article summarization with diverse summary types (Newsroom).
55	openbookqa_0.1.0	Open-book science QA requiring retrieval of core science facts.
56	opinion_abstracts_idebate	Summarizing debate arguments into a single opinion abstract.
57	opinion_abstracts_rotten_tomatoes	Summarizing critic reviews into a consensus opinion abstract.
58	para_crawl_enes	English–Spanish parallel sentence translation (ParaCrawl).
59	paws_wiki_1.1.0	Paraphrase detection with lexical overlap adversaries (PAWS-Wiki).
60	piqa_1.0.0	Physical intuition QA: choosing the correct way to accomplish a task.
61	quac_1.0.0	Question answering in context: QA over Wikipedia sections (QuAC).
62	samsum_1.0.0	Abstractive summarization of messenger-style dialogues (SAMSum).
63	sentiment140_1.0.0	Binary sentiment classification of tweets (Sentiment140).
64	snli_1.1.0	Sentence-pair NLI on image caption premises (Stanford NLI).
65	squad_v1.1_3.0.0	Extractive reading comprehension on Wikipedia passages (SQuAD v1.1).
66	squad_v2.0_3.0.0	Extractive QA with unanswerable questions (SQuAD v2.0).
67	story_cloze_2016_1.0.0	Commonsense reasoning: select the correct ending for a short story.
68	super_glue_cb_1.0.2	Three-way NLI on CommitmentBank short premises (CB).
69	super_glue_copa_1.0.2	Causal reasoning: choose the cause or effect of a given premise (COPA).
70	super_glue_multirc_1.0.2	Multi-sentence reading comprehension with binary answer validation.
71	super_glue_record_1.0.2	Cloze-style QA over news with named-entity answer candidates (ReCoRD).
72	super_glue_rte_1.0.2	Recognizing textual entailment (two-class, SuperGLUE RTE).
73	super_glue_wic_1.0.2	Word-in-context: does a polysemous word share meaning in two sentences?
74	super_glue_wsc.fixed_1.0.2	Winograd Schema Challenge coreference resolution (fixed split).
75	trec_1.0.0	Question type classification into six coarse categories (TREC).
76	trivia_qa_rc_1.1.0	Trivia QA with supporting evidence from Wikipedia/web (reading comprehension).
77	true_case	Restores true casing to all-lowercase input text.
78	unified_qa_science_inst	Unified QA format applied to science exam questions with instructions.
79	winogrande_1.1.0	Large-scale Winograd Schema pronoun resolution at various difficulty levels.
80	word_segment	Word segmentation: insert spaces into a string of concatenated words.
81	yelp_polarity_reviews_0.2.0	Binary sentiment classification of Yelp business reviews.
SGLUE		
82	axb	Broad-coverage NLI diagnostic test on linguistic phenomena (AX-b).
83	axg	Gender-bias NLI diagnostic test (AX-g).
84	boolq	Yes/no QA given a passage (BoolQ, SuperGLUE split).
85	cola	Linguistic acceptability classification (CoLA, SuperGLUE split).
86	copa	Causal commonsense reasoning: cause or effect selection (COPA).
87	cr	Binary sentiment classification of customer product reviews (CR).
88	mpqa	Opinion polarity detection at the phrase level (MPQA).
89	mr	Binary sentiment classification of movie reviews (MR).
90	mrpc	Sentence-pair paraphrase detection (MRPC, SuperGLUE split).
91	pawsx	Cross-lingual paraphrase detection across multiple languages (PAWS-X).
92	qnli	Question-answer NLI (QNLI, SuperGLUE split).
93	qqp	Quora question-pair duplication detection (QQP, SuperGLUE split).

Continued on next page.

Table F2 continued from previous page.

#	Task	Description
94	rte	Recognizing textual entailment (RTE, SuperGLUE split).
95	sst2	Binary sentiment classification of movie review sentences (SST-2).
96	subj	Subjectivity classification: is the sentence subjective or objective?
97	wnli	Winograd NLI coreference-based entailment (WNLI).
98	xwinograd	Cross-lingual Winograd Schema pronoun resolution.
T0		
99	adversarial_qa_dbert_answer_the_following_q	Adversarial QA with DBERT backbone: answer a question given a passage.
100	adversarial_qa_dbert_based_on	Adversarial QA (DBERT): answer framed as "based on the passage..."
101	adversarial_qa_dbert_generate_question	Generate a question whose answer is the given span (DBERT).
102	adversarial_qa_dbert_question_context_answer	QA formatted with explicit question, context, and answer fields (DBERT).
103	adversarial_qa_dbert_tell_what_it_is	Adversarial QA (DBERT): identify what the answer refers to.
104	adversarial_qa_dbidaf_answer_the_following_q	Same as above but using the BiDAF adversarial backbone model.
105	adversarial_qa_dbidaf_based_on	BiDAF adversarial QA with "based on" prompt framing.
106	adversarial_qa_dbidaf_generate_question	Generate a question from an answer span (BiDAF adversarial).
107	adversarial_qa_dbidaf_question_context_answer	Explicit QCA format (BiDAF adversarial).
108	adversarial_qa_dbidaf_tell_what_it_is	Identify the referent of the answer span (BiDAF adversarial).
109	adversarial_qa_droberta_answer_the_following_q	Same as above but using the RoBERTa adversarial backbone model.
110	adversarial_qa_droberta_based_on	RoBERTa adversarial QA with "based on" framing.
111	adversarial_qa_droberta_generate_question	Generate a question from an answer span (RoBERTa adversarial).
112	adversarial_qa_droberta_question_context_answer	Explicit QCA format (RoBERTa adversarial).
113	adversarial_qa_droberta_tell_what_it_is	Identify the referent of the answer span (RoBERTa adversarial).
114	amazon_polarity_Is_this_product_review_positive	Binary product review sentiment: is the review positive?
115	amazon_polarity_Is_this_review	Binary sentiment classification of Amazon product reviews.
116	amazon_polarity_Is_this_review_negative	Binary product review sentiment: is the review negative?
117	amazon_polarity_User_recommend_this_product	Predict whether a user would recommend the product based on review.
118	amazon_polarity_convey_negative_or_positive_sentiment	Classify the sentiment conveyed in an Amazon review.
119	amazon_polarity_flattering_or_not	Determine whether an Amazon review is flattering or not.
120	amazon_polarity_negative_or_positive_tone	Classify the overall tone of an Amazon product review.
121	amazon_polarity_user_satisfied	Predict whether the reviewer is satisfied with the product.
122	amazon_polarity_would_you_buy	Predict purchase intent from a product review.
123	app_reviews_categorize_rating_using_review	Predict a star rating from a mobile app review.
124	app_reviews_convert_to_rating	Convert the sentiment of an app review to a numeric rating.
125	app_reviews_convert_to_star_rating	Convert an app review to an explicit star rating label.
126	app_reviews_generate_review	Generate a plausible app review text given a rating.
127	cos_e_v1.11_aligned_with_common_sense	Commonsense QA: select the answer aligned with common sense (CoS-E).
128	cos_e_v1.11_description_question_option_id	CoS-E formatted with description, question, and option ID.
129	cos_e_v1.11_description_question_option_text	CoS-E formatted with description, question, and option text.
130	cos_e_v1.11_explain_why_human	Generate a human-style explanation for why a commonsense answer is correct.

Continued on next page.

Table F2 continued from previous page.

#	Task	Description
131	cos_e_v1.11_generate_explanation_given_text	Generate a rationale given the question and correct answer.
132	cos_e_v1.11_i_think	CoS-E with first-person "I think. . . " style explanation framing.
133	cos_e_v1.11_question_description_option_id	CoS-E formatted with question, description, and option ID.
134	cos_e_v1.11_question_description_option_text	CoS-E formatted with question, description, and option text.
135	cos_e_v1.11_question_option_description_id	CoS-E formatted with question, option, and description (ID output).
136	cos_e_v1.11_question_option_description_text	CoS-E formatted with question, option, and description (text output).
137	cos_e_v1.11_rationale	Produce the rationale for a CoS-E commonsense answer choice.
138	dbpedia_14_given_a_choice_of_categories_	DBpedia topic classification given a list of category choices.
139	dbpedia_14_given_a_list_of_category_what_does_the_title_belong_to	Predict DBpedia category from article title given a category list.
140	dbpedia_14_given_list_what_category_does_the_paragraph_belong_to	Predict DBpedia category from a paragraph given a category list.
141	dbpedia_14_pick_one_category_for_the_following_text	Pick one DBpedia ontology category for the provided text.
142	dream_answer_to_dialogue	Select the answer to a question about a dialogue (DREAM).
143	dream_baseline	Baseline QA on the DREAM dialogue comprehension dataset.
144	dream_generate_first_utterance	Generate the opening utterance of a dialogue given context.
145	dream_generate_last_utterance	Generate the closing utterance of a dialogue given context.
146	dream_read_the_following_conversation_and_answer_the_question	Standard reading comprehension on DREAM dialogues.
147	duorc_ParaphraseRC_answer_question	Answer a paraphrased question about a movie plot (DuoRC).
148	duorc_ParaphraseRC_build_story_around_qa	Generate a story that supports a given QA pair (DuoRC Paraphrase).
149	duorc_ParaphraseRC_decide_worth_it	Decide whether a QA pair is worth including (DuoRC Paraphrase).
150	duorc_ParaphraseRC_extract_answer	Extract the answer span from a paraphrased plot (DuoRC).
151	duorc_ParaphraseRC_generate_question	Generate a question from a plot and answer (DuoRC Paraphrase).
152	duorc_ParaphraseRC_generate_question_by_answer	Generate a question conditioned solely on the answer (DuoRC Paraphrase).
153	duorc_ParaphraseRC_movie_director	Predict the movie director from a plot description (DuoRC Paraphrase).
154	duorc_ParaphraseRC_question_answering	Standard QA over paraphrased movie plots (DuoRC).
155	duorc_ParaphraseRC_title_generation	Generate a movie title from its plot (DuoRC Paraphrase).
156	duorc_SelfRC_answer_question	Answer a question about a movie plot using the same-source RC (DuoRC SelfRC).
157	duorc_SelfRC_build_story_around_qa	Generate a story supporting a QA pair (DuoRC SelfRC).
158	duorc_SelfRC_decide_worth_it	Decide whether a QA pair is worth including (DuoRC SelfRC).
159	duorc_SelfRC_extract_answer	Extract the answer span from the same-source plot (DuoRC SelfRC).
160	duorc_SelfRC_generate_question	Generate a question from a plot and answer (DuoRC SelfRC).
161	duorc_SelfRC_generate_question_by_answer	Generate a question conditioned solely on the answer (DuoRC SelfRC).
162	duorc_SelfRC_movie_director	Predict the movie director from a same-source plot (DuoRC SelfRC).
163	duorc_SelfRC_question_answering	Standard QA over same-source movie plots (DuoRC SelfRC).
164	duorc_SelfRC_title_generation	Generate a movie title from its same-source plot (DuoRC SelfRC).
165	kilt_tasks_hotpotqa_combining_facts	Multi-hop QA by combining two supporting facts (HotpotQA via KILT).

Continued on next page.

Table F2 continued from previous page.

#	Task	Description
166	kilt_tasks_hotpotqa_complex_question	Answer complex multi-hop questions using Wikipedia (HotpotQA/KILT).
167	kilt_tasks_hotpotqa_final_exam	Multi-hop QA framed as a final-exam question (HotpotQA/KILT).
168	kilt_tasks_hotpotqa_formulate	Formulate an answer from two supporting Wikipedia passages.
169	kilt_tasks_hotpotqa_straighforward_qa	Straightforward QA format over HotpotQA (KILT).
170	qasc_is_correct_1	Verify whether a given fact combination is correct (QASC, format 1).
171	qasc_is_correct_2	Verify whether a given fact combination is correct (QASC, format 2).
172	qasc_qa_with_combined_facts_1	Answer a science question using one pre-combined supporting fact.
173	qasc_qa_with_separated_facts_1	Answer a science question using two separated supporting facts (format 1).
174	qasc_qa_with_separated_facts_2	Answer a science question using two separated supporting facts (format 2).
175	qasc_qa_with_separated_facts_3	Answer a science question using two separated supporting facts (format 3).
176	qasc_qa_with_separated_facts_4	Answer a science question using two separated supporting facts (format 4).
177	qasc_qa_with_separated_facts_5	Answer a science question using two separated supporting facts (format 5).
178	quail_context_description_question_answer_id	Reading comprehension with context, description, question; answer as ID (QUAIL).
179	quail_context_description_question_answer_text	Same as above but answer as text (QUAIL).
180	quail_context_description_question_text	QUAIL reading comprehension: output the answer text given context and description.
181	quail_context_question_answer_description_id	QUAIL format with context, question, answer, and description (ID output).
182	quail_context_question_answer_description_text	QUAIL format with context, question, answer, and description (text output).
183	quail_context_question_description_answer_id	QUAIL: context, question, description; answer as option ID.
184	quail_context_question_description_answer_text	QUAIL: context, question, description; answer as option text.
185	quail_context_question_description_text	QUAIL: generate answer text from context, question, and description.
186	quail_description_context_question_answer_id	QUAIL with description first, then context and question; answer as ID.
187	quail_description_context_question_answer_text	QUAIL with description first; answer as text.
188	quail_description_context_question_text	QUAIL with description-first ordering; free-text answer.
189	quail_no_prompt_id	QUAIL reading comprehension without an explicit prompt template; answer as ID.
190	quail_no_prompt_text	QUAIL reading comprehension without a prompt template; answer as text.
191	quarel_choose_between	Physical comparison QA: choose between two objects based on a scenario (QuaRel).
192	quarel_do_not_use	QuaRel task variant instructing model not to use certain attributes.
193	quarel_heres_a_story	QuaRel framed as a story comprehension question.
194	quarel_logic_test	QuaRel framed as a logic test over qualitative relationships.
195	quarel_testing_students	QuaRel framed as a student-testing scenario.
196	quartz_answer_question_based_on	Answer a science question based on a provided principle (QuaRTz).
197	quartz_answer_question_below	QuaRTz QA with a “answer the question below” instruction framing.
198	quartz_given_the_fact_answer_the_q	QuaRTz: given a qualitative fact, answer the associated question.
199	quartz_having_read_above_passage	QuaRTz QA framed as comprehension after reading the passage above.
200	quartz_paragraph_question_plain_concat	Plain concatenation of paragraph and question for QuaRTz QA.
201	quartz_read_passage_below_choose	QuaRTz: read a passage and choose the correct answer.
202	quartz_use_info_from_paragraph_question	Answer a QuaRTz question using information extracted from the paragraph.
203	quartz_use_info_from_question_paragraph	Answer a QuaRTz question; question presented before the paragraph.
204	quoref_Answer_Friend_Question	Answer a coreference-dependent question as if asked by a friend (Quoref).
205	quoref_Answer_Question_Given_Context	Standard coreference-aware QA given a context passage (Quoref).
206	quoref_Answer_Test	Quoref QA framed as an answer test.

Continued on next page.

Table F2 continued from previous page.

#	Task	Description
207	quoref_Context_Contains_Answer	Verify that the context contains the answer to a coreference question.
208	quoref_Find_Answer	Find the answer span in a passage for a coreference-dependent question.
209	quoref_Found_Context_Online	Quoref QA framed as finding context online to answer a question.
210	quoref_Given_Context_Answer_Question	Quoref: given context, answer the coreference-dependent question.
211	quoref_Guess_Answer	Quoref framed as guessing the answer to a coreference question.
212	quoref_Guess_Title_For_Context	Predict the title of the passage given its coreference-rich content.
213	quoref_Read_And_Extract_	Read a passage and extract the answer to a coreference question.
214	quoref_What_Is_The_Answer	Simple "what is the answer" prompt for Quoref QA.
215	race_high_Is_this_the_right_answer	Verify if a given answer is correct for a high-school reading question (RACE).
216	race_high_Read_the_article_and_answer_the_question_no_option_	High-school RACE QA without explicit answer options.
217	race_high_Select_the_best_answer	High-school RACE multiple-choice reading comprehension.
218	race_high_Select_the_best_answer_generate_span_	RACE (high): select the best answer and generate the answer span.
219	race_high_Select_the_best_answer_no_instructions_	RACE (high) multiple choice without explicit instructions.
220	race_high_Taking_a_test	RACE (high) framed as a student taking an exam.
221	race_high_Write_a_multi_choice_question_for_the_following_article	Generate a multiple-choice question from a high-school article.
222	race_high_Write_a_multi_choice_question_options_given_	Generate a multiple-choice question with pre-given options (RACE high).
223	race_middle_Is_this_the_right_answer	Verify if a given answer is correct for a middle-school reading question (RACE).
224	race_middle_Read_the_article_and_answer_the_question_no_option_	Middle-school RACE QA without explicit answer options.
225	race_middle_Select_the_best_answer	Middle-school RACE multiple-choice reading comprehension.
226	race_middle_Select_the_best_answer_generate_span_	RACE (middle): select the best answer and generate the span.
227	race_middle_Select_the_best_answer_no_instructions_	RACE (middle) multiple choice without explicit instructions.
228	race_middle_Taking_a_test	RACE (middle) framed as a student taking an exam.
229	race_middle_Write_a_multi_choice_question_for_the_following_article	Generate a multiple-choice question from a middle-school article.
230	race_middle_Write_a_multi_choice_question_options_given_	Generate a multiple-choice question with pre-given options (RACE middle).
231	ropes_background_new_situation_answer	QA using background knowledge applied to a new situation (ROPES).
232	ropes_background_situation_middle	ROPES QA with background and situation interleaved.
233	ropes_given_background_situation	ROPES: answer given background knowledge and a specific situation.
234	ropes_new_situation_background_answer	ROPES with new situation presented before the background.
235	ropes_plain_background_situation	Plain concatenation of background and situation for ROPES QA.
236	ropes_plain_bottom_hint	ROPES with a hint placed at the bottom of the prompt.
237	ropes_plain_no_background	ROPES QA without providing the background passage.
238	ropes_prompt_beginning	ROPES with the prompt instruction placed at the beginning.
239	ropes_prompt_bottom_hint_beginning	ROPES with both a bottom hint and a beginning prompt instruction.
240	ropes_prompt_bottom_no_hint	ROPES with prompt at the bottom and no background hint.
241	ropes_prompt_mix	ROPES with a mixed prompt ordering strategy.
242	ropes_read_background_situation	ROPES: read the background and situation, then answer the question.
243	sciq_Direct_Question	Answer a science multiple-choice question directly (SciQ).
244	sciq_Direct_Question_Closed_Book_	Answer a SciQ science question without the supporting passage.
245	sciq_Multiple_Choice	Standard multiple-choice science QA (SciQ).

Continued on next page.

Table F2 continued from previous page.

#	Task	Description
246	sciq_Multiple_Choice_Closed_Book_	Closed-book multiple-choice science QA (SciQ).
247	sciq_Multiple_Choice_Question_First	SciQ with the question presented before the answer options.
248	social_i_qa_Check_if_a_random_answer_is_valid_or_not	Validate whether a random answer is correct for a social scenario (SocialIQA).
249	social_i_qa_Generate_answer	Generate an answer to a social commonsense question (SocialIQA).
250	social_i_qa_Generate_the_question_from_the_answer	Generate the question that matches a given answer (SocialIQA).
251	social_i_qa_I_was_wondering	SocialIQA framed with an "I was wondering. . . " conversational prompt.
252	social_i_qa_Show_choices_and_generate_answer	SocialIQA: present all choices and generate the correct answer.
253	social_i_qa_Show_choices_and_generate_index	SocialIQA: present all choices and output the index of the correct one.
254	web_questions_get_the_answer	Retrieve a short factual answer to a web search question (WebQuestions).
255	web_questions_potential_correct_answer	Generate a potential correct answer for a WebQuestions query.
256	web_questions_question_answer	Standard QA format on the WebQuestions factoid dataset.
257	web_questions_short_general_knowledge_q	Short general-knowledge QA from WebQuestions.
258	web_questions_whats_the_answer	"What's the answer" prompt style for WebQuestions.
259	wiki_bio_comprehension	Comprehension questions about a person based on their Wikipedia biography.
260	wiki_bio_guess_person	Identify the person described in a Wikipedia biography snippet.
261	wiki_bio_key_content	Extract the key content from a Wikipedia biography.
262	wiki_bio_what_content	Describe what content a Wikipedia biography covers.
263	wiki_bio_who	Answer "who is this" given a Wikipedia biography excerpt.
264	wiki_hop_original_choose_best_object_affirmative_1	Multi-hop: choose the best object in an affirmative frame (WikiHop, format 1).
265	wiki_hop_original_choose_best_object_affirmative_2	Multi-hop: choose the best object in an affirmative frame (WikiHop, format 2).
266	wiki_hop_original_choose_best_object_affirmative_3	Multi-hop: choose the best object in an affirmative frame (WikiHop, format 3).
267	wiki_hop_original_choose_best_object_interrogative_1	Multi-hop: choose the best object in an interrogative frame (WikiHop, format 1).
268	wiki_hop_original_choose_best_object_interrogative_2	Multi-hop: choose the best object in an interrogative frame (WikiHop, format 2).
269	wiki_hop_original_explain_relation	Explain the relationship between two entities using Wikipedia (WikiHop).
270	wiki_hop_original_generate_object	Generate the object entity given a subject and relation (WikiHop).
271	wiki_hop_original_generate_subject	Generate the subject entity given an object and relation (WikiHop).
272	wiki_hop_original_generate_subject_and_object	Generate both subject and object entities for a relation (WikiHop).
273	wiki_qa_Decide_good_answer	Decide whether a candidate answer is a good answer (WikiQA).
274	wiki_qa_Direct_Answer_to_Question	Provide a direct answer to a WikiQA factoid question.
275	wiki_qa_Generate_Question_from_Topic	Generate a question given a Wikipedia topic (WikiQA).
276	wiki_qa_Is_This_True_	Verify whether a candidate answer is true for a WikiQA question.
277	wiki_qa_Jeopardy_style	Answer a WikiQA question in Jeopardy-style (answer as a question).
278	wiki_qa_Topic_Prediction_Answer_Only	Predict the Wikipedia topic from the answer alone (WikiQA).
279	wiki_qa_Topic_Prediction_Question_Only	Predict the Wikipedia topic from the question alone (WikiQA).
280	wiki_qa_Topic_Prediction_Question_and_Answer_Pair	Predict the Wikipedia topic given both the question and answer (WikiQA).
281	wiki_qa_automatic_system	WikiQA formatted for automatic IR-based answer selection.
282	wiki_qa_exercise	WikiQA framed as an exercise for answer selection.
283	wiki_qa_found_on_google	WikiQA framed as retrieving an answer found on Google.

Continued on next page.

Table F2 continued from previous page.

#	Task	Description
284	wiqa_does_the_supposed_perturbation_have_an_effect	Classify whether a described perturbation affects a process (WIQA).
285	wiqa_effect_with_label_answer	Predict the effect of a perturbation on a process; output a label (WIQA).
286	wiqa_effect_with_string_answer	Predict the effect of a perturbation on a process; output a string (WIQA).
287	wiqa_what_is_the_final_step_of_the_following_process	Predict the final step of a described scientific process (WIQA).
288	wiqa_what_is_the_missing_first_step	Identify the missing first step in a partially described process (WIQA).
289	wiqa_what_might_be_the_first_step_of_the_process	Generate a plausible first step for a process (WIQA).
290	wiqa_what_might_be_the_last_step_of_the_process	Generate a plausible last step for a process (WIQA).
291	wiqa_which_of_the_following_is_the_supposed_perturbation	Identify which option is the perturbation in a process description (WIQA).
TULU		
292	aya	Multilingual instruction-following data from the Aya initiative.
293	coconot	Instruction-following training data with conflict/contradiction handling.
294	evol_codealpaca	Evolved code instruction data built on Code Alpaca via instruction evolution.
295	flanv2	Instruction tuning data from the FLAN v2 collection.
296	no_robots	High-quality human-written instruction-response pairs (No Robots dataset).
297	numina_math_tir	Mathematical reasoning with tool-integrated reasoning steps (NuminaMath-TIR).
298	openassistant_guanaco	Human preference data from OpenAssistant filtered to the Guanaco subset.
299	sciriff	Scientific instruction-following data across NLP, biology, and chemistry tasks.
300	table_gpt	Instruction data for table understanding and manipulation tasks (TableGPT).
301	tulu_hardcoded	Hardcoded identity and safety responses for the Tulu model.
302	tulu_personas_algebra	Persona-conditioned algebra problem-solving instruction data.
303	tulu_personas_code	Persona-conditioned code generation and debugging instruction data.
304	tulu_personas_if	Persona-conditioned instruction-following data.
305	tulu_personas_math	Persona-conditioned general math instruction data.
306	tulu_personas_math_grade	Persona-conditioned grade-school math instruction data.
307	wildchat	Real user–chatbot conversations from the WildChat corpus.
308	wildguardmix	Safety and refusal training data mixing WildGuard examples.
309	wildjailbreak	Adversarial jailbreak attack and defense instruction data (WildJailbreak).