Empirical Analysis of Task Mixture Effects in Small-scale Instruction Tuning: A Statistical Approach

Jeesu Jung
KAIST
jisu.jung5@gmail.com

Sangkeun Jung*
Chungnam National University
hugmanskj@gmail.com

Abstract

The performance of large language models heavily depends on instruction tuning, especially on task types and mixture ratios. However, previous research has primarily focused on mixing tasks at fixed ratios, lacking a **systematic and quantitative analysis of task-wise interactions across diverse tasks**. Moreover, it has relied heavily on human labeling. To address these limitations, this study conducts empirical experiments on *unlabeled* instruction corpora, varying both the number and proportion of task combinations to identify *effective mixtures*. To minimize manual labeling, we automatically extract five representative tasks—`programming`, `math problem solving`, `history question answering`, `grammar correction`, and `creative writing`—using only a few seed instructions. Across 51 mixtures, we find that 1–2 task mixtures work best with small datasets, while synergistic 3-task mixtures excel with larger data. Task interactions reveal both synergy (e.g., `programming` + `math`) and interference (e.g., `programming` + `creative writing`). These results provide practical guidelines for mixture design tailored to model scale and data size.

1 Introduction

Large Language Models (LLMs) achieve alignment with human intent through the process of instruction tuning (Ouyang et al., 2022; Chung et al., 2024). Instruction tuning enables the model to understand and respond appropriately to a wide range of natural language instructions via supervised learning on high-quality instruction–response pairs. Recent studies have shown that the performance of instruction-tuned models can **significantly** vary depending on the task and format diversity of the training data (Chen et al., 2023; Zhou et al., 2023). Among these factors, *the types of tasks and their mixture proportions* play a crucial role in determining final model performance, sometimes enabling only 1,000 curated examples to reach the performance of over 52,000 unlabeled ones (Zhou et al., 2023), or achieving up to a 5.7× speedup in training (Chen et al., 2023).

Recent work has increasingly investigated how task mixtures affect instruction tuning (Renduchintala et al., 2024), demonstrating that adjusting task proportions can improve overall model quality. However, the approach focuses primarily on mixing tasks at *fixed ratios* and lack a **systematic and quantitative analysis of task-wise interactions across diverse tasks**, leaving important interaction patterns underexplored. Moreover, prior work has relied on human-labeled datasets such as FLAN (Longpre et al., 2023) and P3 (Sanh et al., 2021). These datasets are limited in scope and do not adequately capture scenarios involving diverse and heterogeneous instructions.

In practice, widely-used instruction datasets such as Self-Instruct (Wang et al., 2023b), OpenAssistant (Köpf et al., 2023), and Alpaca (Taori et al., 2023) are composed of model- or human-generated instructions without explicit task labels. Manually annotating task labels in such datasets is costly and time-consuming, and thus impractical at scale. Moreover, understanding inter-task interaction patterns in large-scale corpora remains challenging. This motivates the need for *automatic analysis methods that infer tasks and guide mixture strategies* from **unlabeled instruction data**.

To address this gap, we propose a *minimally-supervised* framework for designing effective task mixtures from unlabeled instruction corpora, along with an analytical methodology for evaluating their impact. While inspired by recent nearest-neighbor sampling approaches (Kou et al., 2024; Chen et al., 2023; Lin et al., 2025), this work represents the first attempt to apply such methods to the study of task mixtures. By systematically varying the number

*Corresponding Author

Findings of the Association for Computational Linguistics: ACL 2026, pages 13168–13186

and proportion of tasks, our goal is to **empirically characterize task-wise interactions** and discover effective combinations. Concretely, our approach consists of the following three stages:

1. **Task Discovery and Data Allocation.** Define a few ($\leq$ 5) seed instructions for 5 representative tasks (`programming`, `math problem solving`, `history QA`, `grammar correction`, `creative writing`) and retrieve relevant data via embedding-based search.

2. **Systematic Mixture Design and Analysis.** Construct all 1–5 task combinations with four training sizes (250, 500, 750, 1,000), yielding 51 mixtures for empirical analysis of task interactions.

3. **Large-scale Experimentation and Interaction Analysis.** Tune all mixtures across three model families (Dubey et al., 2024; Team, 2024; Jiang et al., 2023), quantifying synergy and interference, and identify effective mixtures via Pareto frontier analysis.

Results show that 2-task mixtures are effective for small datasets and single-task gains, while 3-task mixtures improve multi-task performance with larger data. We also identify interaction patterns, including synergy (`programming` + `math`) and interference (`programming` + `creative writing`). These insights provide practical guidelines for instruction tuning.

Our main contributions are as follows:

- The first attempt to apply *automatic task identification from unlabeled instructions*, via seed-based retrieval, to task mixture research.
- A *large-scale empirical analysis* quantifying synergy and interference across task mixtures.
- The analysis shows that synergistic mixtures *outperform uniform task distributions* on various benchmark tasks.
- *Practical guidelines for task mixture design*, tailored to model scale and training data size.

2 Task Mixture Construction

We construct task mixtures from unlabeled instruction data in four steps. First, we define a task as a set of instructions with a shared intent and expected response format. Second, we select five representative task categories and retrieve task-relevant examples using seed-based embedding search. Third, we instantiate all non-empty task combinations under multiple allocation ratios and training budgets. Finally, we analyze the resulting mixtures statistically for both target-task optimization and balanced multi-task performance.

2.1 Definition of Task

Instruction datasets include diverse prompts in content, style, and format. We define a *task* as a group of instructions with a shared intent and formats.

While prior work relied on pre-labeled datasets with fixed task definitions and mixing ratios (Renduchintala et al., 2024; Sanh et al., 2021), modern instruction corpora—often user-generated or automatically collected (Taori et al., 2023; Köpf et al., 2023)—lack explicit task boundaries. This motivates the need for task discovery methods that operate without labeled metadata.

2.2 Selection of Representative Tasks

To facilitate a systematic analysis of task interactions, we select five representative tasks grounded in common evaluation benchmarks:

- **`Programming` (`P`)**: Tasks involving code generation and algorithm implementation.
- **`Math problem solving` (`M`)**: Tasks that require logical reasoning and numerical computation.
- **`History QA` (`H`)**: Tasks requiring factual knowledge and information retrieval.
- **`Grammar correction` (`G`)**: Tasks evaluating grammatical accuracy and language editing capabilities.
- **`Creative writing` (`C`)**: Tasks that assess the model's ability to generate creative content such as poems and stories.

These tasks collectively span essential LLM capabilities, including reasoning (`P`, `M`), linguistic generation (`C`, `G`), and factual knowledge (`H`).

2.3 Embedding-based Task-wise Sampling

Modern instruction datasets are often derived from natural user interactions rather than structured task-specific templates. To extract subsets corresponding to each task, we employ an embedding-based sampling method.

2.3.1 Seed Instruction Collection.

The *seed instructions* serve as core queries for embedding-based sampling of task-specific subsets. For each of the five tasks, we manually collect three prototypical instruction patterns, yielding 15 seed instructions in total. The total list is shown in Appendix A.

We embed seed instructions and the full corpus in a shared space and retrieve the top 1,100 candidates per task using cosine similarity, following recent nearest-neighbor sampling approaches (Kou et al., 2024; Chen et al., 2023; Lin et al., 2025). We employ the `text-embedding-3-small` model (OpenAI, 2023) for semantic similarity, with the sampling process as follows:

1. Embed the seed instructions and the full Alpaca dataset using an LLM-based embedder.
2. Compute the centroid (mean embedding) for each task based on its seed instructions.
3. Retrieve the top 220 nearest instructions per centroid (1,100 total per task) using cosine similarity.

From each sampled set, we randomly select 1,000 examples for training and 100 for testing. The data sizes are aligned with the minimum data thresholds proposed in prior work(Zhou et al., 2023).

2.4 Construction of Task Mixtures

We construct all combinations of 1 to 5 tasks, resulting in 31 unique sets: 5 single-task, 10 two-task $\binom{5}{2}$, 10 three-task $\binom{5}{3}$, 5 four-task $\binom{5}{4}$, and 1 five-task mixture. These are used to analyze baseline performance, pairwise interactions, and broader composition effects.

For each task combination, we design both uniform and skewed distributions (e.g., 2:1:1) to yield a total of *51 distinct task mixtures*.

2.4.1 Training Data Allocation.

To reflect practical training constraints, we construct mixtures under four training data sizes: 250, 500, 750, and 1,000 examples. Task allocations follow the specified mixture ratios and are divided in 50-example increments for granularity.

2.5 Statistical Analysis of Task Mixtures

We consider two objectives of instruction tuning: **maximizing a single target task** and **balancing performance across tasks**. To address both, we treat task proportions as design variables and apply statistical mixture analysis to identify effective combinations.

2.5.1 Score Aggregation across LLMs

Because LLMs produce different score distributions due to their inherent biases, we aggregate scores from three structurally diverse LLM judges into a single value. We denote by $y_d(w)$ the raw score of mixture w on task d. We apply **inverse-variance weighting** so that judges whose evaluations are more consistent across instances receive higher weight. This yields an aggregated score $y_{d^\star}(w)$ for each mixture.

2.5.2 Statistical Analysis for Target Task Improvement

We then cast mixture selection as a *best-arm identification* problem (Gabillon et al., 2012). Using bootstrap resampling over instances ($B = 10{,}000$), we compute for each mixture:

- $p_{\text{best}}(w)$: the probability that w is ranked first,
- $\Delta(w)$: the margin between the winner $\hat{w}$ and the runner-up.

We declare a unique winner $\hat{w}$ if

$$p_{\text{best}}(\hat{w}) \geq 0.95 \quad \text{and} \quad \Pr(\Delta(\hat{w}) > \tau) \geq 0.95,$$

with margin threshold $\tau = 0.03$. Otherwise, we report an ε-optimal set. This procedure provides a **statistically certified Top-1 mixture** for each target task and training data size.

2.5.3 Statistical Analysis for Balanced Multi-task Performance

For balanced performance across all tasks, we normalize per-task scores for comparability:

$$\tilde{y}_d(w) = \frac{y_d(w) - \min_{w'} y_d(w')}{\max_{w'} y_d(w') - \min_{w'} y_d(w')},$$

so that $\tilde{y}_d(w) \in [0, 1]$ represents the normalized performance of mixture w on task d.

We summarize each mixture w by two factors:

1. **Quality** (q): the mean normalized score across tasks,

$$q(w) = \frac{1}{D} \sum_{d=1}^{D} \tilde{y}_d(w).$$

2. **Stability** ($1 - d_\infty$): the complement of worst domain regret,

$$d_\infty(w) = \max_d \big(1 - \tilde{y}_d(w)\big).$$

Thus each mixture is represented as a 2D point $(q(w), 1 - d_\infty(w))$. We compute the **Pareto frontier** in this space to identify mixtures that cannot be improved in both dimensions simultaneously. For reporting, we present the optimal Pareto set. For deployment, we optionally scalarize with

$$Score_\lambda(w) = \lambda\, q(w) + (1 - \lambda)\,(1 - d_\infty(w)),$$

to select a single balanced mixture. We set $\lambda = 0.5$ by default to equally weight performance and diversity, representing a neutral trade-off point.

3 Experimental Setup

We investigate how task mixture composition affects instruction tuning across different training data sizes and model architectures. Our analysis addresses the following questions:

1. **Task Discovery:** Can tasks be clustered from unlabeled data with seed-based embeddings?

2. **Training Data Size:** How do effective mixtures change with more training data?

3. **Target Task Focus:** Which mixtures improve specific tasks?

4. **Balanced Performance:** Which mixtures give strong overall results?

5. **Interaction Effects:** Which mixtures show synergy or interference across different experimental setups?

We systematically examine task mixtures across model architectures under consistent settings, and will release all code and configurations upon publication.

3.1 Embedding Search Setup

we use the `text-embedding-3-small` model (OpenAI, 2023) with cosine distance, for task clustering. All embedding parameters follow OpenAI defaults.

3.2 Models

We evaluate three widely used open-source models: **Llama 3.1 8B (Llama)** (Dubey et al., 2024), **Mistral 7B (Mistral)** (Jiang et al., 2023), **Qwen 2.5 7B (Qwen)** (Team, 2024). This selection enables comparison across different architectures and language pretraining styles.

3.3 Training Setup

All models are fine-tuned using LoRA (Hu et al., 2022) on a single L40 GPU. We train for 5 epochs with a learning rate of 2×10^{-5}, weight decay 0.01, and batch size 4. LoRA settings include rank $r = 8$, $\alpha = 32$, and dropout rate 0.1.

3.4 Dataset

We use the Alpaca instruction dataset (Taori et al., 2023) (52k instruction–response pairs), a diverse unlabeled corpus suitable for post-hoc task construction. From five representative task types—`Programming`, `Math problem solving`, `History QA`, `Grammar correction`, and `Creative writing`—we sample 1,000 training and 100 test examples each. Prompts average 13–17 tokens, while responses vary more widely (20–140 tokens). Detailed statistics are provided in Appendix B.

3.5 Evaluation Dataset and Baselines

3.5.1 Evaluation Dataset Construction

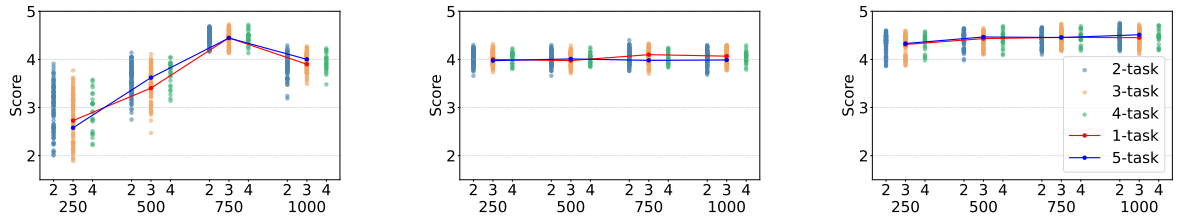
We construct a test set of 500 instructions (100 per task) for evaluation. Scoring follows the LLM-as-a-judge protocol (Zheng et al., 2023), using `GPT-4o` (OpenAI, 2024), Gemini-2.5-flash (et al, 2025b), and Deepseek-chat-v3.1 (et al, 2025a) as the judge models. The prompt is shown in Appendix C.

3.5.2 Baselines

To analyze the influence of task mixtures, we designed two controlled baselines: **Single-task** tuning for target task focus, and **Uniform** training across all five tasks for balanced multi-task. These baselines isolate the effect of task composition without relying on external sampling or weighting schemes.

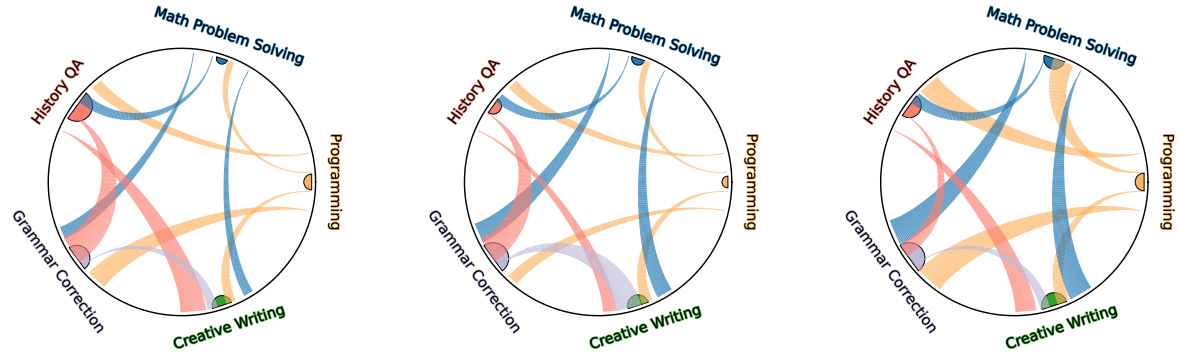
4 Analysis of Task Mixtures

We analyze 51 task mixtures across three model families (Llama, Mistral, Qwen) and four data sizes (250–1,000 examples), addressing: (i) task discovery from unlabeled corpora, (ii) effects of data size,



(a) Llama 3.1 8B (b) Mistral 7B (c) Qwen 2.5 7B

Figure 1: Performance trends by data scale. The blue line represents uniform-task training, while the red line indicates single-task training. Overall, model performance tends to training data size converge between 750 and 1,000. Two-task mixtures show high variance and occasional peaks, while four-task mixtures offer more stability with slightly lower peaks.



(a) Llama 3.1 8B (b) Mistral 7B (c) Qwen 2.5 7B

Figure 2: Contribution relationships among effective mixtures for the Target Task. Edge thickness = frequency of a Task in optimal solutions (over all data sizes); semicircle size = share of self-optimal cases. `Programming(P)` influences all domains, while `Creative writing(C)` never appears in an optimal solution except the self-optimal cases.

	P	M	H	G	C	Avg.
Human 1	0.7	0.9	0.7	1.0	1.0	0.86
Human 2	0.7	0.8	0.8	0.8	0.9	0.80
Human 3	0.7	0.8	0.9	0.8	0.8	0.80
Human 4	0.7	0.7	0.7	0.9	0.8	0.76
Human 5	1.0	0.8	0.9	0.8	1.0	0.90
Majority Voting	0.8	1.0	0.9	0.9	0.9	0.90

Table 1: Agreement rates between human annotators and embedding-searched task assignments, along with majority voting results across task types: `Programming` (P), `Math problem solving` (M), `History QA` (H), `Grammar correction` (G), and `Creative writing` (C). The final column reports the average per annotator.

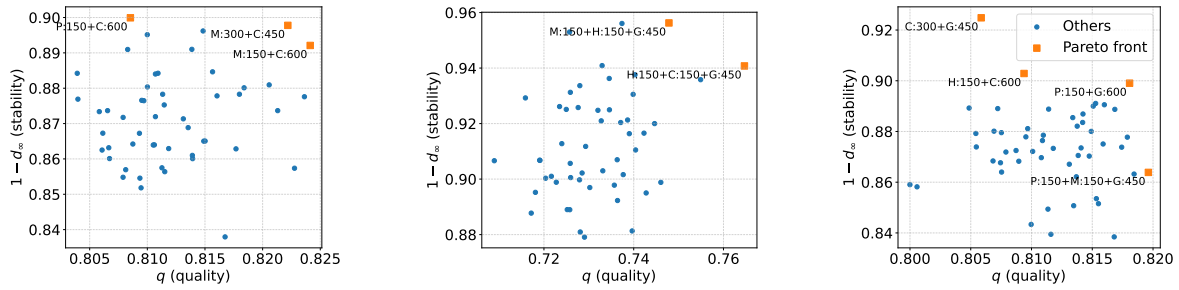
(iii) task-optimal mixtures, (iv) balanced performance, and (v) model-specific differences.

4.1 Reliability of Task Discovery

Table 1 reports human evaluation scores for the embedding-based retrieval procedure. Five annotators were asked to judge whether the retrieved instructions were semantically aligned with the target task domain, using a 0–1 scale where higher values indicate stronger relevance. Across all tasks, individual annotators consistently rated the retrieved samples between 0.7 and 1.0, suggesting that embedding search reliably captures task-related semantics. The average agreement per annotator ranges from 0.76 to 0.90, and majority voting yields an overall score of 0.90, further confirming robustness across evaluators. In particular, `programming` and `creative writing` show perfect or near-perfect agreement, while `history QA` and `grammar correction` exhibit slightly lower but still strong alignment. These results validate our use of embedding-based retrieval as a practical and accurate method for constructing task-specific subsets without manual annotation.

4.2 Effect of Training Data Size

Figure 1 illustrates how performance variance across mixtures decreases as data size increases,



(a) Llama 3.1 8B (b) Mistral 7B (c) Qwen 2.5 7B

Figure 3: Pareto-optimal solutions on overall tasks with 750 data size. Orange squares indicate the optimal solution for each model. The number of data samples per domain is also provided. d_{inf} denotes Stability ($1 - d_\infty$), i.e., the complement of the worst-domain regret, and q represents the average performance across all tasks: `Programming` (`P`), `Math problem solving` (`M`), `History QA` (`H`), `Grammar correction` (`G`), and `Creative writing` (`C`). The final column reports the average score per annotator.

with a convergence point near 750 examples.

- **250–500 examples (low-resource):** single-task or 2-task mixtures are most effective, while 3+ tasks often underperform.
- **750–1,000 examples (mid- to high-resource):** 3–4 task mixtures become more stable and outperform the uniform 5-task baseline in balanced settings.

These results support the strategy of *specialization under low-resource and balanced mixtures under larger data sizes*.

4.3 Target Task Optimization

As defined in Sec. 2.5.2, we treat mixture selection as a best-arm identification problem and determine the statistically significant winner mixture. Specifically, a mixture is declared the winner when its probability of being ranked first. A mixture is declared the unique winner if $p_{\text{best}} \geq 0.95$ and $\Pr[\Delta > \tau] \geq 0.95$. The threshold $\tau = 0.03$ ensures that the declared winner exhibits not only statistical but also practical significance, preventing overinterpretation of marginal gains.

4.3.1 Unique Winners.

Out of 60 (model, size, task) settings, 45 (75%) satisfy these criteria. Varying the p_{best} threshold from 0.90 to 0.975 does not change this count, indicating *threshold-insensitivity*. The remaining 15 settings are reported as ϵ-optimal sets; Appendix D.2 lists the top-1 candidate mixtures and their probabilities.

4.3.2 Domain-specific Patterns.

Figure 2 shows the relation between domain and training data combination.

- `Programming`: Helps other tasks, showing strong synergy on `M` or `G`.
- `Math problem solving`: benefits from `P` as auxiliaries.
- `History QA`: effective alone or in `P` and `M` combinations.
- `Grammar correction`: best when isolated or paired with `H`.
- `Creative writing`: sensitive to interference, often degraded when mixed with `G` or `H`.

4.4 Balanced Multi-task Performance

Having examined task-specific optimization, we now turn to the question of overall balance across tasks. To this end, we analyze task mixtures under the Pareto frontier defined in the $(q, 1 - d_\infty)$ space. We report optimal mixtures on main document, and hypervolume (HV) as a scalar summary and the composition of balanced winners on Appendix D.4.

4.4.1 Pareto Frontier Size.

Most settings yield a single balanced winner. However, for Mistral at 1,000 examples, the Pareto set expands to two distinct mixtures, suggesting the presence of *multiple near-equivalent balanced solutions* rather than a single dominant one.

Winner Mixtures. Balanced winners often emphasize `H` *and* `G` at smaller data sizes (e.g., Llama: `H+G` on 250 data size, Mistral: `H+G` on 500 data size), while `P` *participation increases* at larger budgets (e.g., Qwen: `P+M+G` on 750 data size, Llama: `M+H` on 1,000 data size). Figure 3 shows the optimal configuration at a data size of 750. The optimal

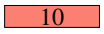
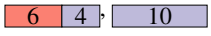
Target task	Best-performing mixture pattern	Harmful pattern
Programming		>40% Math Problem Solving
Math Problem Solving		Math Problem Solving mixed with language-only tasks
History QA		—
Grammar correction		>20% Programming
Creative writing		pure language mixtures

Table 2: Best-performing and harmful mixture patterns for each target task, aggregated over all three models. Bar segments represent component tasks: = Programming (P), = Math problem solving (M), = History QA (H), = Grammar correction (G), = Creative writing (C).

configurations at other data sizes can be found in Appendix D.3. This shift aligns with the specialization → balance transition observed in Sec 4.3.

4.5 Model-wise Differences

Robustness patterns differ across models:

- **Llama:** sensitive to data sizes size, balanced HV decreases with larger data.
- **Mistral:** balanced HV peaks at medium data sizes (500 examples).
- **Qwen:** consistently robust across data sizes, achieving high HV even with 250 examples.

4.6 Take-aways

1. **Convergence point:** mixture effects stabilize near 750 examples.
2. **Unique winner stability:** decisions are threshold-insensitive; $\Pr[\Delta > \tau]$ is the decisive factor.
3. **Balanced performance:** HV trajectories show model-specific peaks—Mistral at 500, Qwen at 250/750, Llama at 250.
4. **Domain/model specificity:** P–M synergize, P–C interfere. Qwen is robust under low-resource, Llama is size-sensitive, Mistral peaks at medium data sizes.

5 Practical Guidelines

Our analysis offers practical guidelines for building instruction-tuning mixtures from unlabeled data, helping practitioners curate training data efficiently.

1. **Data size-aware strategy.** With fewer than 500 examples, single-task or carefully chosen 2-task mixtures are most effective. Beyond 750 examples, balanced mixtures of 3–4 tasks become advantageous. Practitioners should *specialize under low-resource settings and balance under larger scales*.
2. **Domain interactions.** Programming and Math consistently exhibit synergy and should be combined when the target requires reasoning or structured outputs. In contrast, Programming and Creative writing often interfere and should be separated unless sufficient data is available to mitigate negative transfer.
3. **Anchor tasks.** History and Grammar frequently serve as stabilizing anchors in balanced winners, particularly at small to mid data sizes. Including at least 20–30% of such language-oriented tasks improves generalization across other domains.
4. **Model-aware design.** Model families differ in their balanced performance peaks: Llama favors smaller data sizes, Mistral peaks at medium data sizes (around 500), while Qwen remains robust across scales. Mixture strategies should be adjusted to model-specific inductive biases.
5. **Uncertainty reporting.** When mixtures cannot be statistically distinguished, we recommend reporting an ϵ-optimal set of top candidates rather than a single winner. This reduces selection bias and provides a more reproducible basis for deployment.

Overall, these guidelines emphasize that mixture design is not one-size-fits-all: effective strategies depend jointly on data size, domain composition, and model architecture. Our framework identifies such trade-offs without task labels (Table 2).

6 Performance on External Benchmarks

The best mixture identified in this study also performs well on external benchmarks, humaneval (Chen et al., 2021), GSM8K (Cobbe et al., 2021), Creative Writing Benchmark v3(Paech,

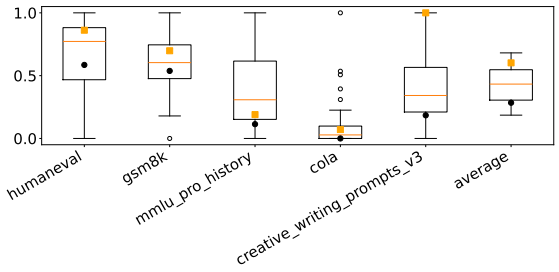


Figure 4: Task-wise benchmark min–max normalized performance and average distribution for Llama trained with 750 examples; orange squares show each task's and balanced mixture's optima, black squares the equal-ratio baseline.

2024), CoLA(Warstadt et al., 2019) and MMLU-pro(Wang et al., 2024b) history. Figure 4 shows its performance on the Llama model trained with 750 examples, the most stable data size. While weaker on creative domains such as `History QA` and `Creative writing`, it excels in overall performance and objective metrics, **achieving the best results for balanced multi-task training**. Since each benchmark uses different scales, we applied min–max normalization. The external benchmark results of other models are provided in Appendix D.5.

7 Related Work

Instruction tuning datasets were initially constructed by aggregating multi-task corpora (Longpre et al., 2023; Sanh et al., 2021) or leveraging large-scale user-generated data (Ouyang et al., 2022; Chung et al., 2024). More recent work emphasizes smaller curated sets (Zhou et al., 2023; Chen et al., 2023) and automated sampling such as embedding-based retrieval (Chen et al., 2025), LLM-based filtering (Liu et al., 2024), and diversity-aware selection (Bukharin et al., 2024).

Another line of research examines how task mixtures affect multi-task tuning. Studies have proposed adaptive reweighting methods (Renduchintala et al., 2024), empirical comparisons of mixing strategies (Wang et al., 2024a; Mueller et al., 2024), and label-free tuning frameworks like InstructZero (Chen et al., 2024), collectively demonstrating the complexity of task interactions.

Statistical mixture modeling has a long history in fields such as chemistry and hyperparameter optimization (Scheffé, 1958; Ehrgott, 2012; Gabillon et al., 2012), but has rarely been applied at the data level in NLP. Our work extends this perspective by treating task proportions as explicit variables for statistical analysis.

Despite these advances, most prior studies assume access to pre-labeled task annotations (Renduchintala et al., 2024; Wang et al., 2024a; Mueller et al., 2024). This assumption limits applicability in real-world unlabeled corpora and hampers scalability when new tasks arise. We address these limitations by proposing an automated framework that constructs effective task mixtures without relying on explicit task labels.

Recent instruction tuning studies often use large models as evaluators (Li et al., 2023; Wang et al., 2023a; Chen et al., 2024), which correlate with human ratings but may introduce bias when evaluation depends on a single judge model. To mitigate this issue, we adopt a multi-judge protocol with three distinct LLMs: GPT-4o (OpenAI, 2024), Gemini (et al, 2025b), and DeepSeek (et al, 2025a). Each mixture is scored independently by all three judges on a 1–5 Likert scale, and the scores are aggregated via inverse-variance weighting so that more consistent judges receive larger weights. We additionally report bootstrap confidence intervals to quantify uncertainty. More broadly, our framework is related to recent work on semantic data selection and embedding-based retrieval (Chen et al., 2025), but differs in that it uses retrieved subsets as approximate task proxies for mixture construction and statistical analysis rather than as fixed training sets. This approximation is supported by human validation of the retrieval stage, where annotator-level agreement falls in the 0.7–0.9 range and majority voting reaches 0.90, while the main claims of our study do not depend on perfect task recovery so long as the retrieved subsets preserve meaningful variation across instruction types.

8 Conclusion and Future Work

We present a method for automatically identifying tasks from unlabeled instruction data and conduct a large-scale empirical analysis of task mixtures. Our results show that synergistic mixtures consistently outperform uniform task distributions across benchmarks, and that mixture composition has measurable, interpretable effects on performance. In particular, we observe consistent synergy between reasoning-oriented tasks (e.g., `Programming` + `Math problem solving`) and interference between linguistically divergent ones (e.g., `Programming` + `Creative writing`).

More broadly, we provide a principled and reproducible framework for studying inter-task dynamics in instruction tuning. By formulating mixture design as a statistical optimization problem, our approach moves beyond heuristic task selection toward interpretable analysis, without relying on task labels.

Our findings reveal both expected and less intuitive patterns. While some align with prior expectations—such as positive interactions between programming and math tasks, and the effectiveness of simpler mixtures in low-resource settings—others do not, including the stabilizing role of language-oriented tasks in balanced mixtures and the limited benefit of increasing task diversity. These results highlight the need for systematic statistical analysis, since intuition alone cannot determine which interaction patterns are robust, scale-dependent, or model-specific.

Future work should test whether these patterns hold with more reasoning-intensive instruction data, substantially larger models, and hybrid human–LLM evaluation. Such extensions will help distinguish general properties of instruction tuning from effects specific to the controlled setting studied here. In this sense, our contribution is both a set of empirical findings and a framework for making these distinctions explicit and testable.

Limitations

Our study intentionally focuses on a controlled and interpretable setting, which also defines its main limitations.

- **Data scope:** We restrict our analysis to direct-answer instruction data in order to maintain relatively consistent task boundaries and support reliable statistical modeling. This choice is motivated by the prevalence of single-turn instruction data in practical instruction tuning. However, the resulting findings should not be assumed to transfer directly to reasoning-intensive settings such as chain-of-thought supervision or to specialized domains with substantially different data distributions.

- **Model scale:** All experiments are conducted on 7–8B parameter models so that 51 task mixtures can be evaluated under a tractable and fully replicated setup. This design enables broad coverage of mixture conditions, but it also limits the extent to which the observed patterns can be generalized to substantially larger models. Some effects identified here may reflect stable trends, while others may change as model capacity increases. Verifying which interaction patterns persist across scale remains an important direction for future work.

- **Evaluation protocol:** Our evaluation uses three LLM judges with bootstrap-based uncertainty estimation, providing a practical balance between cost, reproducibility, and statistical robustness. Although this protocol follows recent practice and reduces dependence on a single evaluator, it does not replace direct human assessment. Additional human evaluation would be valuable, especially for validating subtle interaction effects and for determining whether the observed gains are meaningful beyond model-based preference judgments.

These design choices prioritize controlled statistical analysis over exhaustive coverage. Future work can extend the present framework along three main directions: broader instruction regimes including reasoning-oriented data, larger model scales, and hybrid evaluation protocols combining LLM judges with human assessment. More generally, our current results should be interpreted as evidence about mixture effects in a constrained but practically relevant setting, rather than as a universal characterization of instruction tuning behavior.

Acknowledgments

This work was supported by the IITP grant funded by the Korea government (MSIT) (No. RS-2024-00445087, Enhancing AI Model Reliability Through Domain-Specific Automated Value Alignment Assessment) and (No. RS-2025-25464461, AI's Vision of Harmony: A Fair and Transparent Multimodal Agentic Platform for Conflict Mediation), the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2019-II190004, Development of Semi-Supervised Learning Language Intelligence Technology and Korean Tutoring Service for Foreigners), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2025-00556217314820926401 01), and a grant (26212MFDS008) from the Ministry of Food and Drug Safety in 2026.

References

Alexander Bukharin, Shiyang Li, Zhengyang Wang, Jingfeng Yang, Bing Yin, Xian Li, Chao Zhang, Tuo Zhao, and Haoming Jiang. 2024. Data diversity matters for robust instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3411–3425, Miami, Florida, USA. Association for Computational Linguistics.

Lichang Chen, Jiuhai Chen, Tom Goldstein, Heng Huang, and Tianyi Zhou. 2024. Instructzero: efficient instruction optimization for black-box large language models6518. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2023. Alpagasus: Train a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Yicheng Chen, Yining Li, Kai Hu, Zerun Ma, Haochen Ye, and Kai Chen. 2025. Mig: Automatic data selection for instruction tuning by maximizing information gain in semantic space. In *Findings of the Association for Computational Linguistics: ACL 2025*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tai, 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, and 16 others. 2024. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25(1).

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Ankur Dubey, Arnav Jauhri, Abhishek Pandey, Abhinav Kadian, Ahmad Al-Dahle, and et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Matthias Ehrgott. 2012. Vilfredo pareto and multi-objective optimization. In *Documenta Mathematica Extra Volume ISMP*, pages 447–453. EMIS. Historical review of Pareto optimality and frontier concept.

DeepSeek-AI et al. 2025a. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Gheorghe Comanici et al. 2025b. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. 2012. Best arm identification: A unified approach to fixed budget and fixed confidence. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

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

A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations - democratizing large language model alignment. In *Advances in Neural Information Processing Systems*, volume 36, pages 47669–47681. Curran Associates, Inc.

Jianshang Kou, Benfeng Xu, Chiwei Zhu, and Zhendong Mao. 2024. KNN-instruct: Automatic instruction construction with k nearest neighbor deduction. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10337–10350, Miami, Florida, USA. Association for Computational Linguistics.

Yujia Li, Chenxin Zhang, Zhiruo Wu, and 1 others. 2023. Tuna: Instruction tuning using feedback from large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

I-Fan Lin, Faegheh Hasibi, and Suzan Verberne. 2025. Spill: Domain-adaptive intent clustering based on selection and pooling with large language models. *Preprint*, arXiv:2503.15351.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*.

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 *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR.

David Mueller, Mark Dredze, and Nicholas Andrews. 2024. Multi-task transfer matters during instruction-tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14880–14891, Bangkok, Thailand. Association for Computational Linguistics.

OpenAI. 2023. Openai text-embedding-3-small model. https://platform.openai.com/docs/guides/embeddings. Accessed: 2025-07-22.

OpenAI. 2024. Gpt-4o technical report. Accessed: 2025-07-28.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Samuel J. Paech. 2024. Eq-bench: An emotional intelligence benchmark for large language models.

H S V N S 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, Bangkok, Thailand. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, and 22 others. 2021. Multitask prompted training enables zero-shot task generalization. *Preprint*, arXiv:2110.08207.

Henry Scheffé. 1958. Experiments with mixtures. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):344–360.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Boxin Wang, Shichao Ren, and 1 others. 2023a. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. *arXiv preprint arXiv:2306.05087*.

Renxi Wang, Haonan Li, Minghao Wu, Yuxia Wang, Xudong Han, Chiyu Zhang, and Timothy Baldwin. 2024a. Demystifying instruction mixing for fine-tuning large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 68–75, Bangkok, Thailand. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024b. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pages 95266–95290. Curran Associates, Inc.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Corpus of Linguistic Acceptability (CoLA).

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. In *Advances in Neural Information Processing Systems*, volume 36, pages 55006–55021. Curran Associates, Inc.

A Seed Instructions

Table 3 and Table 4 list the seed instructions used to retrieve task-specific samples. The centroid of these seed sentences was used as the golden key for each task, and the top-k most similar samples were retrieved based on similarity.

B Data Statistics

Table 5 shows the statistics of dataset construction. Samples were collected for five representative benchmark tasks. Overall, the number of input tokens was similar across tasks, ranging from 12 to

Task	Instruction Examples
`Programming`	• Write code to print "hello world". • Develop a strategy for optimizing web content • List three use cases for gpt-3. • Create a function to check if a given number is an armstrong number or not. • Explain what the following machine learning algorithm does. INPUT: k-means clustering
`Math problem solving`	• What is the tenth term in the geometric sequence 9,3,1,1/3, . . . ? • Explain the concept of vector multiplication • Compare and contrast the mass of the earth with the mass of mars. INPUT: earth mass: 5.972 $\times 10^{24}$ kg mars mass: 6.39 $\times 10^{23}$ kg • Convert the phrase "1/2" into a fraction. • Compute the result of 3^8.

Table 3: Seed instructions for `Programming` (`P`) and `Math problem solving` (`M`) task category.

16, while the number of output tokens varied considerably. In particular, the `grammar correction` task exhibited notably shorter output lengths.

C Evaluation Prompt

Table 6 shows the prompt of llm-as-a-judge (Zheng et al., 2023). We referred to the 1–5 scale criteria used in existing benchmark evaluations. The final result was obtained by averaging the evaluation scores from three LLMs.

D Overall Performance and Analysis Results

D.1 Full Results.

Due to the scale of the evaluation—51 task mixtures $\times$ 3 model architectures $\times$ 5 target tasks $\times$ 4 training data sizes—it is impractical to include all detailed results in the main paper or appendix. Instead, we provide the complete results as supplementary material. Specifically, the per-model results are available in the `score` folder as `llama.csv`, `mistral.csv`, and `qwen.csv`, respectively.

D.2 Target Task Optimization

Table 7, 8, 9 shows the multiple optimal candidates for each domain and model. At this time, different results are observed for each model. In the case of Qwen, it presents single optimal case in all cases, but for Llama, the number of optimal candidates increases as the data size grows. Mistral, regardless of data size, shows multiple optimal candidates in the `programming` domain.

D.3 Pareto Frontier Results for Total Data Scale

Pareto frontier results across all data scales are summarized in Table 10. Overall, we observe that mixtures emphasizing language-oriented tasks often yield optimal performance. Figure 5 presents the optimal mixture for each model.

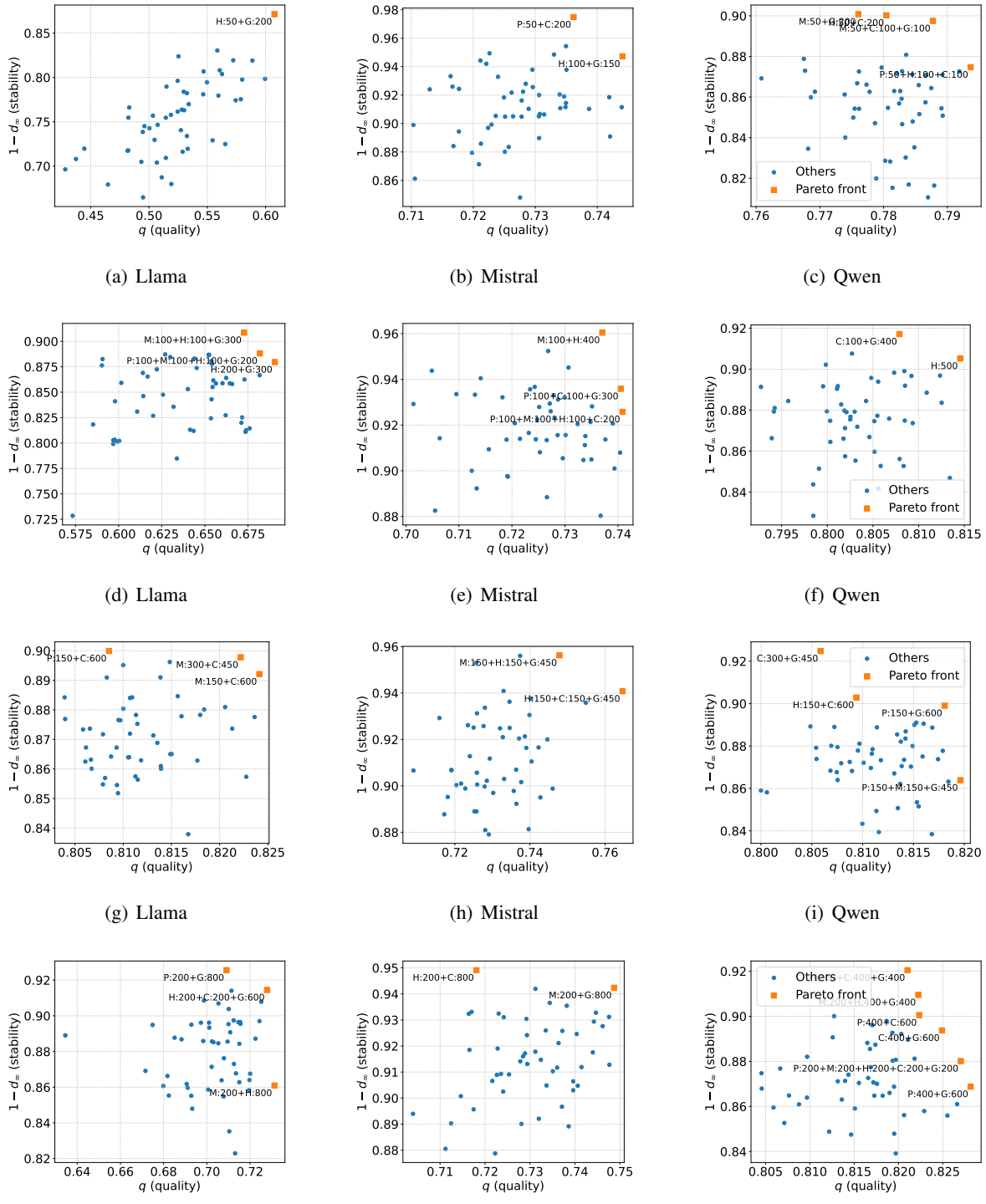
D.4 Hypervolume Trends.

Figure 6 shows HV trajectories per model across data sizes.

- **Llama:** achieves its highest HV at 250 examples (0.968), followed by a gradual decline with larger data sizes (0.900 at 1,000).
- **Mistral:** peaks at 500 examples (0.918), drops sharply at 750 (0.853), and stabilizes at 1,000 (0.859).
- **Qwen:** robust across scales, high at 250 (0.963), dips at 500 (0.920), then rises again at 750 (0.960), and slightly declines at 1,000 (0.922).

Task	Instruction Examples
History QA	• When did the Second World War end? • Summarize the history of the peloponnesian war in no more than 100 words. • List the states in the united states of america in alphabetical order. • Answer the question based on common sense and your knowledge. INPUT: what is the name of the capital city of peru? • Write a biography of bill gates
Grammar correction	• Format the input sentence by adding punctuation. INPUT: i love to eat pizza • Generate a sentence using the following words in the correct order. INPUT: house on fire • Rewrite the sentence using antonyms to two of the adjectives. INPUT: the students found the exam relatively easy. • Rearrange the phrase such that it is in an alphabetical order • Edit the following sentence in the most natural way to make it active voice: INPUT: the media campaign was led by the team.
Creative Writing	• Convert the following text into an acrostic poem. INPUT: time • Generate a children's story with the following title: the magic violin. • Give me a metaphor for a good relationship. • Take a joke and explain it in one sentence. INPUT: why don't scientists trust atoms? because they make up everything. • Categorize the following text into one of the genres: comedy, thriller, romance, or drama. INPUT: two kids from very different backgrounds meet in summer camp and form an unlikely friendship.

Table 4: Seed instructions for History QA (H), Grammar correction (G), and Creative writing (C) task category.



(a) Llama (b) Mistral (c) Qwen

(d) Llama (e) Mistral (f) Qwen

(g) Llama (h) Mistral (i) Qwen

(j) Llama (k) Mistral (l) Qwen

Figure 5: Pareto-optimal solutions on overall tasks with 250, 500, 750, 1000 data size. Orange squares indicate the optimal solution for each model. The number of data samples per domain is also provided. d_{inf} denotes Stability ($1 - d_{\infty}$), i.e., the complement of the worst-domain regret, and q represents the average performance across all tasks: `Programming` (`P`), `Math problem solving` (`M`), `History QA` (`H`), `Grammar correction` (`G`), and `Creative writing` (`C`). The final column reports the average score per annotator.

Source	# of samples	Input	Output
Train			
Programming	1,000	14.2	138.5
Math problem solving	1,000	12.8	86.9
History QA	1,000	14.3	111.3
Grammar correction	1,000	16.4	25.3
Creative writing	1,000	14.6	106.7
Test			
Programming	100	13.7	123.7
Math problem solving	100	12.7	92.3
History QA	100	17.6	117.5
Grammar correction	100	17.1	21.4
Creative writing	100	14.7	88.4

Table 5: Data size and average length (in tokens) of prompts (Input) and responses (Output) across training and test splits. Training data are constructed under varying training data sizes (250 to 1,000 sequences), with task combinations allocated accordingly.

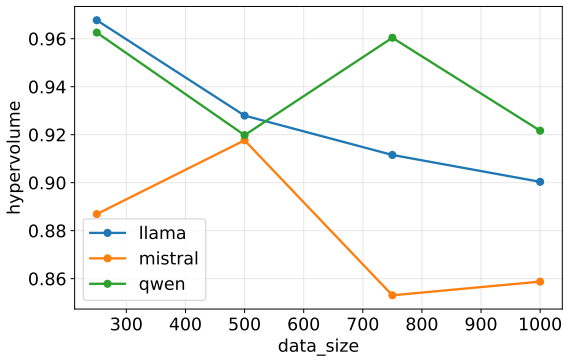
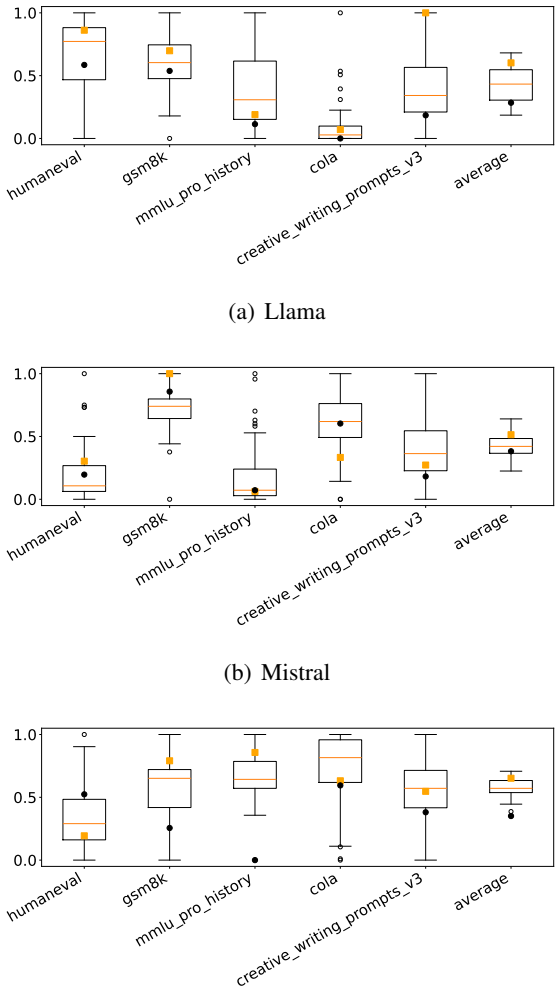


Figure 6: Pareto hypervolume across total data sizes. Higher hypervolume indicates a better Pareto front; Llama shows a consistent decline, Mistral peaks at mid-scale, and Qwen remains robust with a resurgence at 750 examples.



(a) Llama

(b) Mistral

(c) Qwen

Figure 7: Task-wise benchmark min–max normalized performance and average distribution for Llama, Mistral, and Qwen trained with 750 examples; orange squares mark the optima for each task and the balanced multi-task mixture, while black squares indicate the baseline (training all tasks in equal proportion).

These differences indicate that *balanced performance peaks at different data sizes depending on the model*.

D.5 External Benchmark Performance

Figure 7 shows the performance distribution of the Llama, Mistral, and Qwen models with 750 data size on external benchmarks. Although their distributions differ, models selected as the best generally achieve above-average performance. In particular, for average performance, models identified on the Pareto frontier demonstrate superior results across all models. All distributions were computed after applying min–max normalization.

Evaluation Prompt

You are an impartial evaluator.
Your task is to assign a score from 1 to 5
for the response to the given question,
based on four criteria:
accuracy, relevance, completeness, and fluency.

Scoring Guide:
5 – Excellent: Fully accurate and relevant;
answers the question completely with detail;
language is natural and well-structured.
4 – Good: Mostly accurate and relevant;
answers the question but lacks minor details or clarity;
language is mostly fluent.
3 – Fair: Partially accurate or relevant;
incomplete or vague explanation; some awkward phrasing.
2 – Poor: Largely inaccurate or off-topic;
misinterprets question; hard to follow.
1 – Very Poor: Completely incorrect or irrelevant;
nonsensical or no meaningful answer.

Format your answer strictly as: [[X]]
Where X is the score from 1 to 5. Do not add anything else.

—

Question: question

Response: response

Score:

Table 6: Evaluation prompt for judging scores.

Data size	Target task	Best mixture	p_{best}	$\Pr[\Delta > \tau]$_eff
250	P	H:250	1.000	1.000
	M	H:100+G:150	1.000	0.999
	H	H:250	1.000	1.000
	G	H:50+G:200	1.000	1.000
	C	H:50+G:200	1.000	1.000
500	P	P:100+H:100+G:300	0.957	0.882
	M	H:500	0.889	0.787
	H	H:200+C:300	1.000	0.995
	G	M:100+H:100+C:100+G:200	0.521	0.335
	C	H:200+C:300	0.999	0.996
750	P	P:750	1.000	1.000
	M	P:300+G:450	1.000	0.994
	H	H:300+C:450	1.000	1.000
	G	P:150+C:150+G:450	1.000	1.000
	C	M:150+C:600	1.000	1.000
1000	P	P:200+M:200+G:600	0.990	0.946
	M	M:200+H:800	0.983	0.945
	H	P:400+H:600	0.678	0.450
	G	M:200+H:200+G:600	1.000	1.000
	C	P:200+C:800	0.991	0.959

Table 7: Optimal mixture of Llama for the 250-, 500-, 750- and 1,000-examples settings across five tasks: Programming (P), Math problem solving (M), History QA (H), Grammar correction (G), and Creative writing (C). We report p_{best}, $\Pr[\Delta > \tau]$, and the top-1 candidate indices with selection probabilities.

Data size	Target task	Best mixture	p_{best}	$\Pr[\Delta > \tau]$_eff
250	P	M:50+C:50+G:150	0.973	0.886
	M	P:50+M:50+C:50+G:100	0.971	0.895
	H	H:50+C:100+G:100	0.969	0.875
	G	P:50+C:200	0.990	0.972
	C	H:100+G:150	1.000	1.000
500	P	M:100+G:400	0.712	0.448
	M	P:100+M:100+H:100+C:200	0.988	0.931
	H	H:200+G:300	0.926	0.793
	G	P:200+G:300	0.997	0.992
	C	M:100+G:400	1.000	1.000
750	P	C:750	0.977	0.872
	M	P:300+H:450	0.992	0.955
	H	M:150+H:600	0.992	0.966
	G	H:150+C:150+G:450	1.000	0.999
	C	H:150+C:150+G:450	1.000	1.000
1000	P	M:200+H:200+C:600	0.992	0.939
	M	M:200+G:800	0.991	0.954
	H	H:200+C:200+G:600	0.984	0.925
	G	C:200+G:800	1.000	0.998
	C	M:400+C:600	1.000	1.000

Table 8: Optimal mixture of Mistral for the 250-, 500-, 750- and 1,000-examples settings across five tasks: Programming (P), Math problem solving (M), History QA (H), Grammar correction (G), and Creative writing (C). We report p_{best}, $\Pr[\Delta > \tau]$, and the top-1 candidate indices with selection probabilities.

Data size	Target task	Best mixture	p_{best}	$\Pr[\Delta > \tau]$_eff
250	P	P:50+H:200	1.0	0.999
	M	M:50+C:200	1.0	0.999
	H	P:50+M:100+C:100	1.0	1.000
	G	H:50+C:200	1.0	1.000
	C	P:50+M:50+H:50+G:100	1.0	1.000
500	P	M:100+G:400	1.0	1.000
	M	G:500	1.0	0.999
	H	M:100+H:100+G:300	1.0	1.000
	G	H:500	1.0	1.000
	C	P:100+G:400	1.0	1.000
750	P	P:150+M:150+G:450	1.0	1.000
	M	P:300+G:450	1.0	0.997
	H	H:300+C:450	1.0	1.000
	G	P:150+M:300+H:300	1.0	1.000
	C	P:150+C:600	1.0	1.000
1000	P	P:400+H:600	1.0	0.999
	M	P:200+M:400+G:400	1.0	1.000
	H	P:200+M:800	1.0	1.000
	G	H:200+C:400+G:400	1.0	1.000
	C	M:400+C:600	1.0	1.000

Table 9: Optimal mixture of Qwen for the 250-, 500-, 750- and 1,000-examples settings across five tasks: Programming (P), Math problem solving (M), History QA (H), Grammar correction (G), and Creative writing (C). We report p_{best}, $\Pr[\Delta > \tau]$, and the top-1 candidate indices with selection probabilities.

Model	Samples	Best Mixture	Mean Score	d_∞	M + P ratio
Llama	250	H:G(2:8)	3.382	0.21	0
	500	H:G(4:6)	3.834	0.23	0
	750	H:C(2:8)	4.482	0.12	0
	1,000	H:C:G(2:2:6)	4.054	0.19	0
Mistral	250	H:G(4:6)	4.088	0.14	0
	500	P:C:G(2:2:6)	4.114	0.09	0.2
	750	H:C:G(2:2:6)	4.222	0.09	0
	1,000	M:H(2:8)	4.116	0.16	0.2
Qwen	250	H(1)	4.356	0.10	0
	500	H(1)	4.486	0.10	0
	750	P:G(4:6)	4.494	0.10	0.4
	1,000	P:M:H:C:G(2:2:2:2:2)	4.512	0.10	0.4

Table 10: Pareto frontier results for the 250-, 500-, 750- and 1,000-examples settings across five tasks: Programming (P), Math problem solving (M), History QA (H), Grammar correction (G), and Creative writing (C). Mixtures emphasizing linguistic tasks often dominate. d_∞ measures the maximum gap from the per-task optimum; lower values indicate more balanced performance.