

Journal of Data Science, Statistics, and Visualisation

June 2026, Volume VI, Issue IV.

doi: 10.52933/jdssv.v6i4.171

StatLLaMA: Multi-Stage Training for Domain-Optimized Statistical Large Language Models

Jing-Yi Zeng

National Yang Ming Chiao Tung University

Guan-Hua Huang

National Yang Ming Chiao Tung University

Abstract

This study investigates how to efficiently build a domain-specialized large language model (LLM) for statistics using the lightweight LLaMA-3.2-3B family as the foundation model (FM). We systematically compare three multi-stage training pipelines—starting from a base FM with no instruction-following capability, a base FM augmented with post-hoc instruction tuning, and an instruction-tuned FM with strong general reasoning abilities—across continual pretraining, supervised fine-tuning (SFT), and reinforcement learning from human feedback (RLHF) preference alignment. Results show that pipelines beginning with a base FM fail to develop meaningful statistical reasoning, even after extensive instruction tuning, SFT, or RLHF alignment. In contrast, starting from LLaMA-3.2-3B-Instruct enables effective domain specialization. A comprehensive evaluation of SFT variants reveals clear trade-offs between domain expertise and general reasoning ability. We further demonstrate that direct preference optimization provides stable and effective RLHF preference alignment. The final model, StatLLaMA, achieves strong and balanced performance on benchmarks of mathematical reasoning, common-sense reasoning, and statistical expertise, offering a practical blueprint for developing resource-efficient statistical LLMs. The code is available at <https://github.com/HuangDLab/StatLLaMA>.

Keywords: Foundation model, supervised fine-tuning, continual pretraining, instruction tuning, reinforcement learning from human feedback.

1. Introduction

Large language models (LLMs) based on Transformer architectures have become a central tool in modern natural language processing. Models such as BERT (Devlin et al. 2019), GPT (Radford et al. 2018, 2019; Brown et al. 2020), and the LLaMA family (Touvron et al. 2023a,b; Grattafiori et al. 2024) have demonstrated strong performance across a wide range of general-purpose language tasks, including reasoning, summarization, and question answering. Trained on massive text corpora, these foundation models exhibit broad linguistic competence and extensive general knowledge. However, their effectiveness in specialized technical domains—particularly statistics—remains limited.

Statistical reasoning requires precise use of domain-specific terminology, adherence to formal definitions, and careful multi-step analytical reasoning. In practice, general-purpose LLMs often produce explanations that are shallow, imprecise, or conceptually incorrect when applied to statistical tasks. This gap raises an important applied question: how can statistical knowledge and reasoning be systematically integrated into LLMs without degrading their general language and reasoning abilities? This study addresses this question by developing and evaluating a resource-efficient training framework for constructing a statistical LLM. We focus on the lightweight LLaMA-3.2-3B model, motivated by the practical need for deployable models under constrained computational resources. Our objective is to build a model that achieves strong performance on statistical reasoning tasks while maintaining robust general-purpose capabilities.

Supervised fine-tuning (SFT) is the standard approach for adapting pretrained language models to downstream tasks. However, full fine-tuning becomes prohibitively expensive for large models and is inefficient when multiple domain-specific variants are required. Parameter-efficient fine-tuning (PEFT) methods, particularly low-rank adaptation (LoRA) (Hu et al. 2021), address this limitation by updating only a small subset of parameters while keeping the base model fixed. These techniques substantially reduce computational cost and are well suited for domain adaptation. Nonetheless, PEFT alone does not guarantee effective internalization of complex domain knowledge.

Continual pretraining (CoP) (Gupta et al. 2023; Jin et al. 2022) offers an additional mechanism for domain adaptation by exposing models to large volumes of unlabeled, domain-specific text. While CoP can improve familiarity with statistical language and terminology, our experiments indicate that it provides limited gains for task-oriented statistical reasoning when applied to models without strong instruction-following capabilities. In contrast, instruction-tuned models—trained explicitly to interpret and execute natural-language instructions—have shown superior generalization and usability in interactive settings (Longpre et al. 2023; Wang et al. 2023; Taori et al. 2023). This distinction motivates an important design choice explored in this study: whether domain adaptation is more effective when starting from a base model or from an instruction-tuned model.

Even after instruction tuning and SFT, alignment with human judgment remains a challenge, particularly for nuanced reasoning tasks. Reinforcement learning from human feedback (RLHF) (Bai et al. 2022) addresses this by optimizing model outputs using preference comparisons, but it is computationally intensive and sensitive to hyperparameter choices. More recent alternatives, including direct preference optimization (DPO) (Rafailov et al. 2023) and group relative policy optimization (GRPO) (Shao et al. 2024), provide simpler and more efficient alignment mechanisms. This study presents, to our knowledge, the first empirical comparison of DPO and GRPO in the context of statistical language modeling. To support preference-based optimization, we generate high-quality paired preference data using a stronger external language model (Gemini) as a teacher, following a knowledge distillation paradigm (Hinton et al. 2015; Calderon et al. 2023). This approach enables scalable preference supervision while maintaining consistent judgment criteria.

Building on these insights, we propose and evaluate a multi-stage training paradigm for developing a domain-specialized statistical LLM. The process begins with the LLaMA-3.2-3B model and incorporates either CoP combined with instruction tuning or SFT using the parameter-efficient LoRA method. These stages are designed to inject core statistical knowledge and task structures into the model. Next, we apply preference alignment using DPO or GRPO to fine-tune the model’s outputs according to human preferences. To assess the effectiveness of this approach, we conducted extensive experiments across several benchmark tasks: GSM8K (Cobbe et al. 2021) for mathematical reasoning, AP Statistics multiple-choice questions (AP Program 2024) for statistical knowledge, and ARC (Clark et al. 2018) for commonsense reasoning.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the datasets and training stages. Section 4 details the experimental design and training pipelines. Section 5 presents quantitative and qualitative results. Section 6 concludes with a discussion of limitations and future directions.

2. Related Work

2.1. Foundation models

Foundation models constitute the backbone of modern natural language processing (NLP). Prominent examples—including BERT (Devlin et al. 2019), GPT (Radford et al. 2018, 2019; Brown et al. 2020), and the LLaMA family (Touvron et al. 2023a,b; Grattafiori et al. 2024)—are based on the Transformer architecture and pretrained on massive text corpora using self-supervised objectives (Jing and Tian 2021). This pretraining endows models with broad linguistic competence, general reasoning ability, and strong transferability across tasks.

Despite their versatility, foundation models are inherently general-purpose. When applied to technical domains such as statistics, they often struggle with precise terminology, formal definitions, and multi-step analytical reasoning. These limitations motivate the need for targeted domain adaptation strategies that enhance domain-specific competence while preserving general reasoning ability. Our work builds on this line of research by adapting the lightweight LLaMA-3.2-3B model for statistical reasoning.

2.2. Supervised fine-tuning

SFT adapts a pretrained language model to downstream tasks using labeled input–output pairs. Given a dataset $\mathcal{D} = \{(x, y)\}$ of the input x and the target output y , model parameters θ are optimized by minimizing the negative log-likelihood loss:

$$L_{\text{SFT}}(\theta) = -\mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\sum_{t=1}^{|y|} \log \pi_{\theta}(y_t | x, y_{<t}) \right],$$

where $\pi_{\theta}(y_t | x, y_{<t})$ denotes the probability assigned by the model to token y_t , given the input x and the preceding tokens $y_{<t}$.

Historically, SFT has been implemented via full fine-tuning (FFT), updating all model parameters. While effective, FFT is computationally expensive for large models and prone to catastrophic forgetting (Kirkpatrick et al. 2017; Lopez-Paz and Ranzato 2017; Huang et al. 2024). Early work such as ULMFiT (Howard and Ruder 2018) introduced strategies—discriminative learning rates, gradual unfreezing, and carefully scheduled learning rates—to mitigate these risks and improve stability. These ideas laid the foundation for more efficient adaptation methods used in modern LLM training.

2.3. Parameter-efficient fine-tuning

PEFT addresses the scalability limitations of FFT by freezing the pretrained parameters and updating only a small set of additional task-specific parameters. This substantially reduces computational cost, storage requirements, and the risk of catastrophic forgetting.

Among PEFT methods, LoRA (Hu et al. 2021) has become particularly influential. LoRA models the weight update to a pretrained matrix $W_0 \in \mathbb{R}^{d \times k}$ as a low-rank decomposition $\Delta W = BA$, where $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$, with rank $r \ll \min(d, k)$. The original weight W_0 remains frozen, A is randomly initialized (e.g., Gaussian), and B is initialized to zero, ensuring that ΔW is initially zero and does not alter the model’s output at the start of training. The adapted weights take the form

$$W' = W_0 + \left(\frac{\alpha}{r} \right) BA,$$

where α controls the adaptation strength. LoRA approach greatly reduces the number of trainable parameters and achieves performance comparable to FFT while introducing negligible inference overhead, making it well suited for deployment in resource-constrained settings.

Extensions such as QLoRA (Detmeters et al. 2023b) further improve efficiency by combining LoRA with low-bit quantization, while other variants explore alternative decompositions (Liu et al. 2024) or layer-wise adaptations (Lialin et al. 2024). In this study, LoRA serves as the primary mechanism for efficient domain adaptation.

2.4. Continual pretraining

CoP extends the original self-supervised training objective by further training a pretrained model on large volumes of domain-specific, typically unlabeled text. By immersing the model in authentic domain language, CoP improves familiarity with specialized terminology, discourse patterns, and background knowledge.

A key advantage of CoP is that it eliminates the need for costly manual annotation. Empirical studies have shown CoP to be effective in terminology-dense or stylistically distinct domains (Gururangan et al. 2020). However, CoP remains fundamentally a language modeling task: it improves sequence prediction and knowledge absorption but does not directly teach task execution, instruction following, or decision-making. As a result, CoP alone is insufficient for application-oriented settings and must be combined with supervised or instruction-based fine-tuning.

2.5. Instruction tuning

Instruction tuning enhances a model’s ability to understand and execute natural language instructions by fine-tuning on datasets formatted as instruction-response pairs. Although the training objective is identical to standard SFT, the instructional structure of the data enables models to generalize across tasks and formats, substantially improving zero-shot and few-shot performance (Wei et al. 2022; Ouyang et al. 2022; Chung et al. 2024).

Instruction tuning is driven largely by the scale, diversity, and quality of its training data. Early work (e.g., FLAN (Longpre et al. 2023), InstructGPT (Ouyang et al. 2022)) reformatted existing NLP datasets into instruction–response pairs. More recent approaches automate data generation: Self-Instruct (Wang et al. 2023) and XL-Instruct (Iyer et al. 2025) use stronger teacher models to synthesize large instruction corpora, reducing annotation cost but risking redundancy and stylistic uniformity. Complementary efforts such as Alpaca (Taori et al. 2023) incorporate real user–model dialogues; although smaller and constrained by privacy, such data capture authentic interactions and demonstrate the feasibility of instruction tuning under limited resources.

2.6. Reinforcement learning from human feedback

While SFT and instruction tuning improve task execution, they do not ensure alignment with nuanced human preferences or quality judgments. RLHF addresses this limitation by learning from pairwise human preferences over model outputs.

A standard RLHF pipeline comprises three stages. First, a pretrained model is fine-tuned on high-quality demonstrations to obtain an initial policy π_{ref} . Second, human annotators rank candidate responses for a prompt x , yielding preference pairs (x, y_w, y_l) , where y_w is preferred to y_l . A reward model RM_ϕ is trained on these pairs, typically via a Bradley-Terry-style objective,

$$\text{loss}(\phi) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [\log \sigma (\text{RM}_\phi(x, y_w) - \text{RM}_\phi(x, y_l))],$$

to approximate human judgment. Third, the language model is optimized as a policy π_θ , using rewards from RM_ϕ to favor preferred responses during generation.

Proximal policy optimization (PPO) (Schulman et al. 2017; Zheng et al. 2023) is the most widely used reinforcement learning algorithm in RLHF. It provides stable updates through clipping and KL regularization but is computationally intensive and sensitive to hyperparameters. To reduce complexity, alternative methods have been proposed. DPO (Rafailov et al. 2023) removes the explicit reward model and directly optimizes a classification-style objective derived from preference pairs. DPO offers improved stability and lower computational cost, though its effectiveness depends on the quality of

the underlying SFT model (Pan et al. 2025). GRPO (Shao et al. 2024), a lightweight PPO variant, estimates advantages using group-level baselines and reduces memory overhead while retaining PPO’s stabilization mechanisms. This study provides an empirical comparison of DPO and GRPO in the context of statistical language modeling.

2.7. Knowledge distillation

Knowledge distillation (KD) transfers information from a large teacher model to a smaller student model (Hinton et al. 2015), enabling efficient deployment without substantial performance loss. In natural language generation, sequence-level KD—training the student on complete outputs generated by the teacher—offers a practical approach when teacher logits are inaccessible (Calderon et al. 2023).

Beyond model compression, KD has become an effective strategy for generating high-quality synthetic supervision. Teacher-generated responses can serve as pseudo-labels for SFT or as preference data for alignment methods, substantially reducing reliance on costly human annotation. In this work, we employ sequence-level KD using a stronger external LLM (Gemini) to construct both supervised and preference datasets, enabling scalable and consistent training signals for statistical domain adaptation.

3. Materials

3.1. Dataset collection

This study used a variety of datasets for developing and evaluating a large language model specialized in statistics. A complete overview of dataset names, sources, descriptions, and sizes is provided in Table 1. To build a strong foundation in statistics, we collected large-scale corpora, primarily for CoP, aiming to inject background knowledge, terminology, and linguistic patterns into the model. For instruction tuning and SFT, we collected and generated diverse instruction/question–answer (QA) datasets. These datasets trained the model to understand and follow instructions, perform QA, and execute specific tasks. To ensure alignment with human preferences, we curated pairwise preference datasets, with portions of both preference and QA data generated through carefully crafted prompts to the application programming interface (API) of Gemini. Finally, model performance was assessed using a suite of benchmark datasets, enabling a multidimensional evaluation of both general and domain-specific capabilities.

3.2. Construction of training datasets

To effectively leverage heterogeneous datasets across multiple training stages, this study adopted a systematic framework for dataset construction and integration. Beyond the cleaning and normalization of raw data (Section 4.1), this framework encompasses tailored data organization schemes, mixing strategies, prompt engineering, and format conversions aligned with specific training objectives, including CoP, instruction tuning, SFT, and RLHF. These design choices aim to maximize data utility, facilitate targeted capability acquisition at each stage, and improve training efficiency under limited computational resources. Detailed configurations and dataset usage across training stages

Table 1: Overview of datasets collected in this study.

Dataset name	Source	Description	Size
Large corpora			
S2ORC	S2ORC	Academic paper paragraphs filtered by statistical keywords	7,496
Instruction/Question – Answer (QA)			
OpenHermes 2.5	Hugging Face	General-purpose instruction-following dataset	100,000
Dolly-15k	Hugging Face	General-purpose instruction-following dataset	15,000
Statistical Nouns/Defs	Wiki	Statistical terms with definitions	1,203
Statistical CoT	Gemini API	Chain-of-thought (CoT) QA pairs in statistics	1,207
FineTome-100k	Hugging Face	High-quality supervised fine-tuning dataset	20,000
Math-QA	Kaggle	Math problem QA dataset	10,000
GSM8K (Train)	OpenAI/Hugging Face	Grade-school math word problems (training set only, for model training)	7,473
Preference data			
Statistical GRPO	Gemini API	Preference pairs for GRPO training	2,255
Statistical DPO	Gemini API	Preference pairs for DPO training	2,382
Math DPO	Hugging Face	Preference pairs for DPO training	2,393
Benchmark			
GSM8K (Test)	OpenAI/Hugging Face	Grade-school math problems (test set reserved for 8-shot CoT evaluation)	1,319
ARC	Allen Institute for AI	Commonsense reasoning benchmark (0-shot evaluation on Easy Set)	2,357
AP Statistics	AP Central	College-level introductory statistics multiple-choice questions	287

are provided in Appendix A.

3.3. Model evaluation benchmarks

To comprehensively evaluate the performance of the statistical LLM developed in this study, an integrated evaluation framework was established. The evaluation system combines quantitative monitoring metrics on standardized benchmark tests and a custom-built domain-specific evaluation dataset, and qualitative expert analysis.

Grade School Math 8K (GSM8K)

GSM8K, released by OpenAI, is a widely used benchmark for assessing multi-step mathematical reasoning. It contains approximately 8,500 natural language math word problems, each paired with a complete step-by-step solution. This study adopted the 8-shot CoT evaluation paradigm (**GSM8K (Test)**), in which the model receives eight annotated examples of “question-reasoning-answer” before being asked a new question. This setup encourages the model to emulate step-by-step logical reasoning during inference. The primary evaluation metric is answer accuracy, reflecting the model’s capability in logical reasoning and numerical computation.

AI2 Reasoning Challenge (ARC)

Developed by the Allen Institute for AI (AI2), the **ARC** benchmark evaluates common-sense reasoning using questions derived from U.S. grade 3-9 science examinations. To investigate whether domain-specific training in statistics affects general reasoning ability, this study conducted a zero-shot evaluation on the ARC-Easy Set, where no examples are provided prior to questioning. By analyzing changes in accuracy on this benchmark, we can determine whether specialized training introduces interference or whether the model maintains its general reasoning competence alongside its domain specialization.

AP Statistics Benchmark

Given the absence of public benchmarks tailored to testing core statistical competence, we constructed a dedicated **AP Statistics** benchmark. The dataset was sourced primarily from publicly available question bank CrackAP.com’s AP Statistics Practice Tests, initially containing 678 multiple-choice questions. After manual curation to remove items containing images or unsuitable formatting, 287 representative text-only questions were retained. All questions were standardized into a uniform format, each consisting of a question stem and five options labeled (A)–(E), and stored in JSON format for reproducibility. For assessment, a custom prompt instructed the model to output only the letter corresponding to the correct choice, with regular expressions to automatically extract the answer from the model output. The final evaluation metric was accuracy, providing a reliable, quantitative measure of statistical domain expertise.

Qualitative evaluation

Beyond quantitative benchmarks, qualitative expert evaluation was conducted to provide a deeper assessment of the model’s conceptual understanding and expressive

precision. We selected three open-ended questions illustrating representative use cases of statistical reasoning: conceptual question-answering, factual querying from knowledge graphs, and multi-turn consultation scenarios. Appendix B includes these representative examples that show the performance differences between different models.

Responses from models at different training stages (e.g., an instruction-tuned baseline versus a statistics-specialized model) were compared and reviewed by human experts. Evaluation focused on conceptual accuracy, logical coherence, clarity of explanation, appropriate use of terminology, and ability to recognize or avoid statistical fallacies. This expert-reviewed qualitative analysis complements quantitative metrics by offering high-resolution insights into the model’s reasoning structure, interpretive accuracy, and linguistic expressiveness.

4. Methods

4.1. Data preprocessing

Transforming raw, heterogeneous data into high-quality input suitable for LLM training is a foundational step in building a domain-specific model. This process demands not only precise technical implementation but also a careful understanding of diverse data characteristics, potential pitfalls, and effective processing strategies. Appendix C outlines the specific practices and technical considerations applied in this study to prepare heterogeneous statistics-related data for training and alignment.

4.2. Model architecture and training framework

Foundation models and parameter-efficient fine-tuning

This study adopts Meta’s LLaMA-3.2-3B as the base foundation model for all fine-tuning and experimental procedures. The model is built on an optimized Transformer decoder-only architecture, featuring several key improvements over the original Transformer design (Ibe 2024). LLaMA-3.2-3B was selected for this study due to its open availability, competitive performance, and excellent compatibility with PEFT methods. PEFT in this study primarily employed LoRA, which introduces low-rank trainable matrices into targeted model layers, enabling efficient adaptation with minimal parameter updates. LoRA’s rank and scaling factor (α) were adjusted dynamically based on task type and training stage. To further reduce memory consumption and computation costs, the base model was loaded using 4-bit quantization via the bitsandbytes library (Dettmers et al. 2023a). Training employed the AdamW optimizer, with gradient accumulation used to achieve an effective batch size appropriate for available hardware resources.

Training and fine-tuning framework

To accommodate both local and cloud-based resource environments, the overall experimental workflow was built on the Hugging Face ecosystem. For small- to medium-scale GPU setups, we employed the Unsloth acceleration library (Han et al. 2023), which is

optimized for PEFT methods such as LoRA. By leveraging optimized CUDA kernels and efficient memory management, Unsloth reduces VRAM usage by approximately 20–30% and increases training speed by 1.5–2× compared with the standard Hugging Face Trainer, based on preliminary testing.

For large-scale training during the CoP stage or for distributed multi-GPU fine-tuning, we integrated the **DeepSpeed** framework (Rasley et al. 2020). Developed by Microsoft, DeepSpeed implements advanced distributed optimization strategies, most notably the ZeRO (zero redundancy optimizer), which shards model parameters, gradients, and optimizer states across multiple GPUs, thereby minimizing memory overhead. This study utilized ZeRO Stage 2 with 8 A100 GPUs provided by Taiwan Computing Cloud, significantly accelerating training throughput. While **DeepSpeed** offers excellent scalability, integration required careful configuration and compatibility tuning with the Hugging Face Trainer, along with additional engineering considerations such as loss monitoring and runtime stability.

4.3. Core experimental method: Multi-stage training strategy

To address the challenges of domain adaptation for LLMs and to establish an efficient framework for developing a high-performance statistical LLM, this study systematically designed and compared three representative multi-stage training pipelines. Each pipeline integrates different sequences and combinations of techniques to examine how training order and stage composition affect model performance and adaptation efficiency.

All pipelines used the constructed training datasets described in Section 3.2, were implemented under the training framework outlined in Section 4.2, and were evaluated using the benchmarks and metrics detailed in Section 3.3.

Pipeline 1 — Knowledge-first strategy

Pipeline 1 (Figure 1) prioritizes early domain knowledge injection. The underlying hypothesis is that exposing the model to a large corpus of statistical texts before any instruction-based fine-tuning allows it to deeply internalize domain-specific concepts, terminology, and discourse structure. This immersion is expected to establish a strong foundation in statistical reasoning and linguistic patterns, thereby supporting later task-oriented fine-tuning.

Implementation begins with the LLaMA-3.2-3B Base model, which undergoes CoP using a mixed statistical corpus. This corpus, comprising academic paper paragraphs (S20RC) and core statistical terminology and definitions (Statistical Nouns/Defs), serves to enhance the model’s grasp of professional vocabulary and conceptual relationships. After CoP, the model proceeds to SFT using structured statistical instruction and question–answer data, allowing it to apply the learned knowledge in interactive, task-based contexts. Finally, DPO version of RLHF is applied to align the model’s outputs with human judgment, leveraging expert-reviewed statistical preference data. This step enhances the clarity, accuracy, and professionalism of responses. Pipeline 1, therefore, represents a “knowledge-first” paradigm, evaluating whether early domain immersion—prior to SFT—can yield stronger specialization and whether it introduces limitations in later task learning performance.

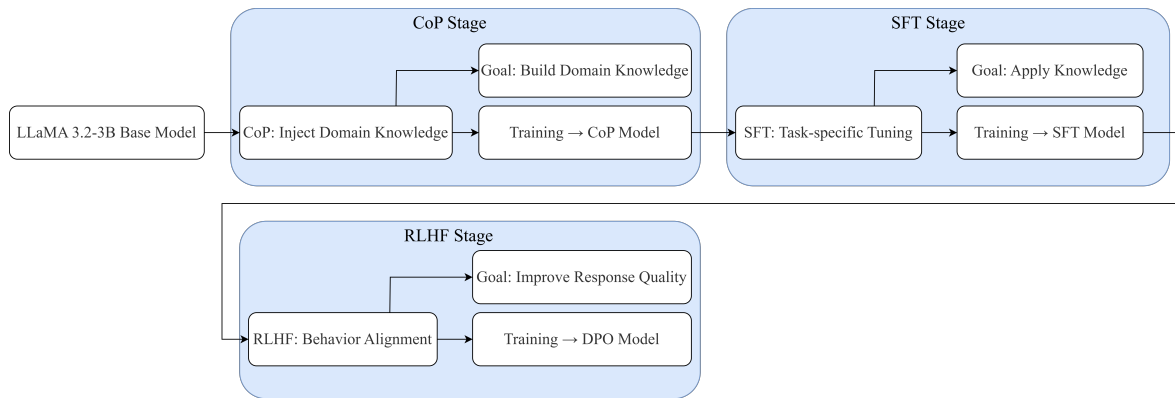


Figure 1: Flowchart of Pipeline 1.

Pipeline 2 — Instruction-bridge strategy

Pipeline 2 (Figure 2) was designed to address a key limitation of Pipeline 1—its potential deficiency in instruction-following ability. While early domain training builds knowledge depth, the model still requires basic interaction and comprehension skills to effectively apply that knowledge. Hence, Pipeline 2 introduces an instruction tuning stage as an intermediary between CoP and SFT.

Implementation begins identically to Pipeline 1, with CoP performed on the same statistical corpus to establish foundational domain knowledge. Next, an instruction tuning phase using open-source, general-purpose instruction datasets (e.g., `Dolly-15k` and `OpenHermes-2.5`) equips the model with broad interaction and command-following capabilities. Subsequently, the model undergoes domain-specific SFT, integrating statistical reasoning, question answering, and task-based learning. The final RLHF-DPO stage further refines the model’s responses, improving fluency and adherence to statistical reasoning norms.

This pipeline investigates the synergistic relationship between early knowledge acquisition and general instruction capability. Specifically, Pipeline 2 tests whether interleaving general instruction tuning before domain-specific fine-tuning leads to superior adaptation and efficiency compared to Pipeline 1. Additionally, it provides a basis for comparison against models that begin with pre-existing instruction-following abilities.

Pipeline 3 — Instruction-first domain adaptation

Pipeline 3 (Figure 3) represents the core experimental direction of this study and diverges fundamentally from the previous two strategies. Instead of starting from a base model, Pipeline 3 builds upon LLaMA-3.2-3B-Instruct, a model that has already undergone large-scale instruction fine-tuning and possesses robust general reasoning and instruction-following abilities. The hypothesis underlying this strategy is that a model with strong general-purpose capabilities may require only targeted domain adaptation to achieve expert-level performance. This could yield comparable or superior results with reduced resource consumption and training time.

Training begins with LLaMA-3.2-3B-Instruct’s SFT on statistical tasks to orient the model toward the domain’s specific concepts, data structures, and reasoning patterns.

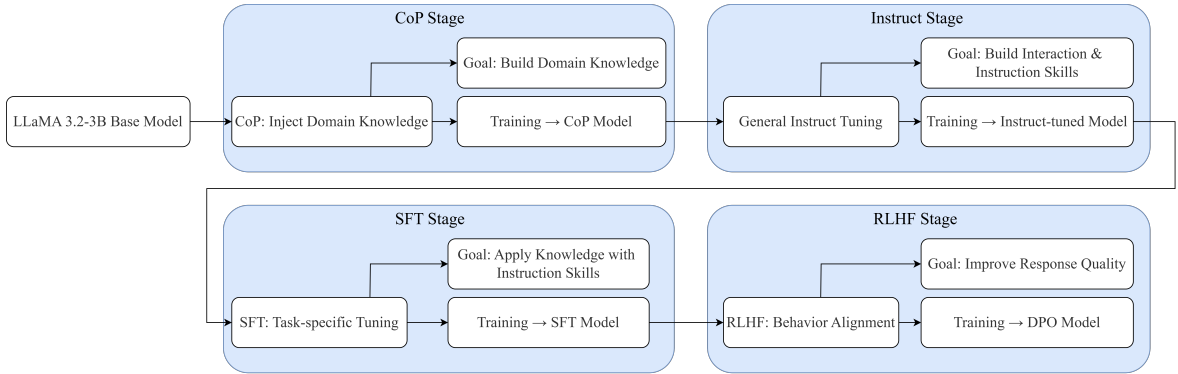


Figure 2: Flowchart of Pipeline 2.

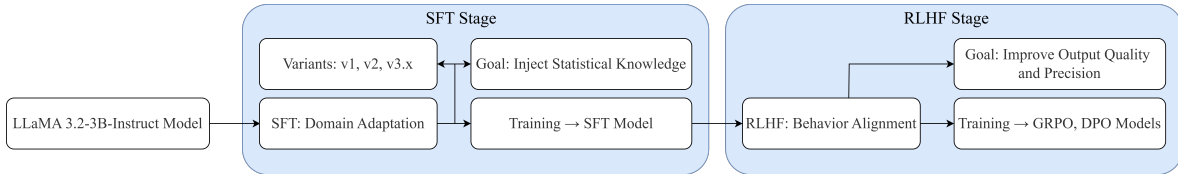


Figure 3: Flowchart of Pipeline 3.

Next, RLHF is used for preference alignment to enhance response quality and domain accuracy. Within the SFT stage, multiple sub-versions were implemented to examine how different configurations—such as data composition, fine-tuning length, and parameter-efficient tuning strategies (e.g., training only response tokens)—affect performance. Similarly, RLHF preference optimization methods were compared between GRPO and DPO.

Pipeline 3 thus aims to validate the feasibility and efficiency of domain adaptation based on a high-performance instruction model. Through comparative analysis with Pipelines 1 and 2, it evaluates trade-offs in training cost, domain knowledge retention, and task effectiveness, providing insights into optimal strategies for building specialized LLMs under resource constraints.

5. Experimental Results

This section presents and analyzes the results of the three multi-stage training pipelines introduced in Section 4.3. The overarching goal is to evaluate how different combinations of knowledge injection, instruction tuning, supervised fine-tuning, and preference optimization contribute to constructing a high-performance statistical LLM. A key focus is determining how to effectively integrate statistical-domain knowledge into the LLaMA-3.2-3B architecture while maintaining—or even enhancing—its general reasoning capabilities.

To establish a clear baseline, we begin by reporting the performance of the LLaMA-3.2-3B Base model and its Instruct variant on the three core evaluation benchmarks: **GSM8K (Test)** (8-shot + CoT), **AP Statistics** (0-shot), and **ARC** (0-shot).

- Base—accuracy %: 29.87 (GSM8K (Test)), 26.13 (AP Statistics), 42.24 (ARC)
- Instruct—accuracy %: 64.44 (GSM8K (Test)), 37.63 (AP Statistics), 43.60 (ARC)

As expected, instruction tuning yields significant gains in mathematical reasoning, general reasoning, and instruction following. These baseline results serve as essential reference points for interpreting the effectiveness of the three training pipelines.

5.1. Pipeline 1: Limitations of direct domain knowledge injection

Pipeline 1 begins with the LLaMA-3.2-3B Base model and prioritizes domain-specific CoP, operating under the hypothesis that early exposure to large amounts of statistics text would establish a strong knowledge foundation. The benchmark results are shown in Table 2.

Table 2: Benchmark results for training Pipeline 1 (knowledge-first strategy).

Model	GSM8K (Test) (8-shot + CoT, Acc. %)	AP Statistics (0-shot, Acc. %)	ARC (0-shot, Acc. %)
LLaMA-3.2-3B	29.87	26.13	42.24
+ CoP	29.49	26.13	42.24
+ CoP + SFT	33.89	26.13	43.52
+ CoP + SFT + RLHF	27.90	26.48	43.34

Values in boldface denote the best result for each benchmark.

The results indicate that Pipeline 1 is largely ineffective. Despite extensive exposure to statistics text, CoP produced no meaningful improvement on any benchmark. All scores remained near baseline. This suggests that passive knowledge exposure—without task-guided objectives—does not lead to improved statistical reasoning or conceptual understanding. The model gains shallow familiarity with specialized terminology but cannot effectively map the newly acquired domain knowledge to interactive task formats.

SFT yields a modest improvement on GSM8K (Test) (+4.4%) and ARC (+1.28%), likely due to the introduction of structured QA data. However, the AP Statistics score remains unchanged, indicating that the model fails to translate knowledge acquired during CoP and SFT into effective domain-specific problem solving.

The RLHF stage unexpectedly damages performance: GSM8K (Test) drops sharply to 27.90%, and AP Statistics remains stagnant. The most plausible explanation is that the Base model lacks sufficient instruction-following capability to interpret preference signals. Without a reliable understanding of prompts or response structure, the model cannot effectively learn the semantic differences between preferred and rejected responses (y_w vs. y_l), resulting in noisy or misdirected optimization.

In summary, Pipeline 1 demonstrates that early knowledge injection without underlying instruction-following capability is ineffective. Domain knowledge alone is insufficient for task-level improvement and is difficult to leverage during subsequent stages.

5.2. Pipeline 2: Adding instruction tuning to bridge knowledge and task ability

Pipeline 2 attempts to address the weaknesses of Pipeline 1 by inserting a general-purpose instruction-tuning stage between CoP and SFT. The intent is to strengthen the model’s ability to interpret tasks and interact meaningfully with SFT and preference data. Results are shown in Table 3.

Table 3: Benchmark results for training Pipeline 2 (knowledge + instruction-bridge strategy).

Model	GSM8K (Test) (8-shot + CoT, Acc. %)	AP Statistics (0-shot, Acc. %)	ARC (0-shot, Acc. %)
LLaMA-3.2-3B + CoP	29.49	26.13	42.24
+ Instruct	26.69	25.78	42.06
+ Instruct + SFT	36.47	25.78	43.52
+ Instruct + SFT + RLHF	26.99	26.13	43.17

Values in boldface denote the best result for each benchmark.

After instruction tuning, performance on all benchmarks declines slightly relative to the CoP stage. This transient drop is expected, as the model adapts to a new objective—learning to interpret and follow instructions—which can temporarily disrupt previously learned patterns. Nonetheless, this stage is essential because it prepares the model to more effectively utilize subsequent SFT data.

The benefit of instruction tuning becomes evident in the SFT stage: **GSM8K (Test)** improves substantially to 36.47%, exceeding the corresponding result in Pipeline 1 (33.89%). This supports the hypothesis that instruction-following capability enhances the effectiveness of SFT. However, **AP Statistics** shows no improvement and in fact falls below its no-instruction-tuned counterpart (26.13%), indicating that domain knowledge from CoP and SFT still fails to transfer to statistical reasoning tasks. This pattern suggests the presence of an “alignment tax,” where introducing instruction tuning between CoP and SFT may interfere with domain-specific learning.

Consistent with Pipeline 1, RLHF further degrades performance: **GSM8K (Test)** drops to 26.99%, **AP Statistics** exhibits only minor fluctuations, and **ARC** declines slightly to 43.17%. Despite improved instruction-following ability, the model appears to lack a well-integrated knowledge structure, leading to unstable or ineffective preference optimization.

To better interpret these negative results, we include a diagnostic analysis (Appendix D) that isolates the contribution of CoP. Removing the CoP stage from Pipelines 1 and 2 yields nearly identical performance across all benchmarks (Table 7), indicating that CoP provides little to no measurable benefit in this setting. A likely explanation is that the CoP datasets (**S20RC** and **Statistical Nouns/Defs**) substantially overlap with the original pretraining corpus of LLaMA-3.2-3B, resulting in minimal new information gain.

In summary, Pipeline 2 yields only modest gains over Pipeline 1—primarily on **GSM8K (Test)**—while failing to improve statistical reasoning. It exhibits weak integration of domain knowledge and unstable behavior during preference optimization. These

limitations likely reflect an alignment tax introduced by inserting instruction tuning between CoP and SFT, as well as redundancy in the CoP datasets. Overall, the results indicate that augmenting instruction-following capability after CoP is substantially less effective than initializing from a model with strong instruction abilities, a limitation that Pipeline 3 is designed to address.

5.3. Pipeline 3: An efficient path for domain adaptation and optimization

Pipeline 3 constitutes the primary experimental focus of this study. Its central strategy is to leverage the strong general language understanding and instruction-following capabilities of LLaMA-3.2-3B-Instruct as an efficient foundation for specialization in the statistics domain. Building on this starting point, we systematically evaluated how different SFT strategies, data integration schemes, and preference optimization methods (GRPO versus DPO) interact to shape the final model performance.

SFT stage: Design and comparison of multi-version strategies (v1, v2, v3.x)

The objective of the SFT stage in Pipeline 3 was to adapt the general-purpose capabilities of LLaMA-3.2-3B-Instruct toward statistics-focused reasoning and expression. To this end, we designed three distinct SFT strategies, each reflecting a different philosophy of training structure and data mixing. Detailed descriptions of all SFT variants, along with their corresponding benchmark results, are provided in Appendix E.1.

Overall, the SFT results demonstrate that no single strategy is universally optimal; instead, model selection depends on the desired balance between general reasoning ability and statistical expertise. SFT-v2 offers a strong and relatively balanced baseline, while variants v3.3 and v3.4 achieve higher statistical performance at the cost of some degradation in general reasoning. Based on these trade-offs, SFT-v2, SFT-v3.3, and SFT-v3.4 were selected as candidates for subsequent preference optimization.

RLHF preference optimization: From GRPO limitations to effective DPO alignment

Following SFT, we proceeded to the preference optimization stage, aiming to further improve response quality, coherence, and alignment with human preferences beyond what can be achieved through SFT alone. We evaluated two methods—GRPO and DPO—and report results for both. While DPO ultimately proved more effective, we treat the GRPO findings not merely as a negative result but as a diagnostic contribution: the observed failure modes illuminate the structural conditions under which group-relative policy optimization is likely to struggle in technical domain adaptation, providing practical guidance for future practitioners.

GRPO experiments and diagnostic analysis. We applied GRPO to the SFT-v1, SFT-v2, and SFT-v3.4 models across three hyperparameter configurations varying LoRA rank, alpha, and the data mixing ratio between **Statistical GRPO** and **GSM8K (Train)** samples (Table 10, Appendix E.2). Benchmark results are reported in Table 11. Across all configurations, GRPO exhibited high sensitivity to hyperparameter

settings and consistently redistributed performance across benchmarks rather than producing uniform improvements. In several cases, modest domain gains were accompanied by substantial degradation in general reasoning. Due to this instability and limited controllability, we concluded that GRPO was not suitable as the primary alignment method in our setting.

Three interacting factors explain this pattern, each specific to or exacerbated by the technical statistical reasoning setting. First, GRPO estimates advantages using group-level baselines computed over a set of sampled responses for each prompt (Shao et al. 2024). In a narrow technical domain such as statistics, the diversity of plausible responses to a given prompt is inherently limited: correct answers tend to converge on the same formal reasoning chain, while incorrect answers cluster around a small set of common misconceptions. This low within-group response diversity yields noisy, low-variance advantage estimates, which in turn produce unstable gradient updates. This contrasts with more open-ended domains, such as general conversation or creative writing, where response diversity is naturally higher and group-relative signals are more informative. Second, our statistical preference dataset (Statistical GRPO) contains 2,255 samples, which is modest relative to the complexity of the optimization landscape that GRPO must navigate. GRPO’s group-relative mechanism is more sensitive to dataset size than DPO’s pairwise classification objective, because each gradient update in GRPO depends on within-batch response comparison rather than on fixed preference pairs. With limited data, the signal-to-noise ratio of the group-relative advantage degrades further. Third, GRPO retains PPO-style policy update mechanics, including clipping and KL regularization, which introduce additional hyperparameter dependencies. In our setting, the interaction between LoRA rank, KL penalty weight, and the Statistical GRPO-to-GSM8K data mixing ratio created a sensitive optimization surface.

DPO experiments. We therefore shifted our focus to DPO, which reframes preference learning as a direct classification-style optimization on paired preference data, avoiding explicit reward modeling and complex reinforcement learning updates. Theoretically, this offers a more stable and straightforward training process.

We applied DPO to three key SFT models: SFT-v2 (balanced capabilities), SFT-v3.3 and SFT-v3.4 (stronger AP Statistics but weakened general performance). The DPO results are summarized in Table 4.

DPO exhibits a much more favorable profile. For SFT-v3.3 and SFT-v3.4, DPO substantially restores general abilities (GSM8K (Test), ARC) that had been weakened during SFT while preserving or slightly improving AP Statistics. SFT-v3.4 + DPO provides the strongest overall performance, with improvements across GSM8K (Test) and ARC and a slight AP Statistics gain. For SFT-v2, DPO maintains AP Statistics and ARC roughly at their original level while slightly reducing GSM8K (Test), yielding a relatively balanced but not dominant configuration compared to SFT-v3.4 + DPO.

Mechanistically, several factors likely contribute to DPO’s advantage in our setting. First, DPO’s classification-style objective operates directly on paired preference data without requiring a separately trained reward model or explicit policy rollouts, which reduces the number of sources of instability. Second, as noted by Pan et al. (2025), DPO’s effectiveness is strongly conditioned on the quality of the underlying SFT

Table 4: Benchmark results from the DPO preference optimization stage used with SFT-v2 and SFT-v3.x in Pipeline 3.

Model	GSM8K (Test) (8-shot + CoT, Acc. %)	AP Statistics (0-shot, Acc. %)	ARC (0-shot, Acc. %)
LLaMA-3.2-3B-Instruct	64.44	37.63	43.60
+ SFT-v2	58.53	39.72	40.36
+ SFT-v2 + DPO	57.85	40.07	40.27
+ SFT-v3.3	53.60	41.46	36.77
+ SFT-v3.3 + DPO	57.24	41.46	41.13
+ SFT-v3.4	54.59	41.11	40.61
+ SFT-v3.4 + DPO	58.98	41.46	41.81

Values in boldface denote the best result for each benchmark.

model—a condition that Pipeline 3 satisfies by virtue of starting from a well-initialized instruction model.

These results highlight DPO’s balancing power: it can restore or improve general reasoning while maintaining domain expertise, rather than trading one off against the other. Among all configurations, SFT-v3.4 + DPO emerges as the most balanced and highest-performing model across the three benchmarks. We designate this final model as StatLLaMA. In the subsequent qualitative analysis, we focus on examining StatLLaMA’s behavior and responses in depth.

5.4. Qualitative analysis: In-depth examination of model capabilities

In addition to the quantitative benchmarks, we conducted a qualitative assessment to further examine the statistical proficiency and interactive behavior of the proposed model. This analysis focuses on the extent to which the final model, StatLLaMA, exhibits improved conceptual accuracy, reasoning depth, and adherence to task constraints relative to its initialization model, LLaMA-3.2-3B-Instruct.

We evaluated three representative use cases: (i) standard conceptual question answering, (ii) knowledge-graph-based factual querying grounded in the ontological structure, and (iii) multi-turn consultation scenarios that approximate applied statistical reasoning. For each case, we constructed prompts targeting domain-relevant constructs, including foundational inferential concepts, ontology-based entity relations, and practical data-analytic decision making. Responses were independently reviewed by researchers with statistical training using the following criteria: conceptual correctness, logical coherence, clarity of exposition, terminological precision, mitigation of common misconceptions, and appropriateness of interaction flow. These representative examples are provided in Appendix B.

Across the three scenarios, StatLLaMA demonstrated notably improved performance. In the conceptual question-answering task (Appendix B Table 5, Prompt 1), its explanation of the distinction between descriptive and inferential statistics was more structured and aligned with standard pedagogical formulations, and the example invoking

confidence intervals accurately illustrated inferential reasoning. In the knowledge-graph query task (Appendix B Table 5, Prompt 2), StatLLaMA adhered more closely to the requested schema, producing responses that remained within the specified ontology and avoided extraneous information—an error exhibited by the baseline model. In the consulting scenario (Appendix B Table 5, Prompt 3), StatLLaMA provided recommendations consistent with accepted practice, including identifying repeated-measures ANOVA as an appropriate method, articulating relevant assumptions such as sphericity, and suggesting mixed-effects modeling as an alternative. These responses indicate a higher level of procedural and conceptual integration. They also provide partial evidences that StatLLaMA’s responses were well-grounded in formal definitions and not critically compromised by teacher-model distillation.

Taken together, the qualitative evidence complements the quantitative results reported earlier. StatLLaMA exhibits improved domain alignment, more coherent reasoning, and more context-appropriate interaction behavior. The examples in Appendix B Table 5 illustrate how multi-stage training introduced in this study leads to measurable gains in applied statistical competence.

6. Conclusion

6.1. Summary of findings

This study evaluated multi-stage training strategies for constructing a statistical large language model that balances domain-specific expertise with general reasoning ability. Using the lightweight LLaMA-3.2-3B family, we targeted a model suitable for deployment in resource-constrained environments while maintaining strong performance on statistical tasks.

Across three training pipelines, we find that the choice of starting model is decisive. Pipelines initialized from a base model lacking instruction-following capability show limited capacity to absorb statistical knowledge, even after extensive CoP and subsequent instruction tuning. In these settings, later SFT or preference-alignment stage yields weak or unstable gains.

By contrast, Pipeline 3—initialized from LLaMA-3.2-3B-Instruct—achieves substantially higher efficiency and effectiveness. Starting from an instruction-capable model enables SFT to specialize the model toward statistical reasoning. Systematic comparison of SFT variants (v1, v2, v3.x) reveals a clear trade-off: aggressive domain specialization improves statistical performance but may degrade general reasoning on benchmarks such as **GSM8K (Test)** and **ARC**. The v3.x series quantifies this balance and supports principled model selection.

For preference alignment, DPO consistently outperforms GRPO. GRPO exhibits high sensitivity to hyperparameters and unstable cross-metric trade-offs. DPO, in contrast, reliably improves response quality, aligns outputs with statistical norms, and restores general reasoning capacity diminished during SFT. The combination of SFT-v3.4 followed by DPO yields the most balanced and robust model.

Based on these results, the combination LLaMA-3.2-3B-Instruct + SFT-v3.4 + DPO was identified as optimal. This final model, **StatLLaMA**, achieves strong and stable performance across **AP Statistics**, **GSM8K (Test)**, and **ARC**. More broadly, the study establishes an effective workflow for building lightweight, domain-specialized LLMs:

High-quality Instruct initialization → Domain-oriented SFT → Stable DPO preference alignment.

6.2. Limitations

Several limitations merit discussion. First, all experiments were conducted at the 3B parameter scale. While appropriate for constrained deployment, it remains unclear whether the observed training dynamics generalize to larger models, which may exhibit different capacity, stability, and knowledge-absorption properties. Results reported at this scale should therefore be interpreted with appropriate caution before extrapolating to broader model families.

Second, the evaluation suite does not fully capture applied statistical practice. Although **AP Statistics**, **GSM8K**, and **ARC** were selected to represent complementary reasoning dimensions—domain expertise, multi-step mathematical reasoning, and general commonsense reasoning, respectively—they do not assess end-to-end analytical workflows, robustness to noisy or ambiguous inputs, or detection of methodological misuse. The breadth of statistical competence that would be required for real-world deployment exceeds what these three benchmarks can collectively measure.

Third, the current study reports results from single training runs for each pipeline and model configuration. Given the known sources of variance in LLM training, including random parameter initialization, data shuffling, and stochastic gradient updates, single-run results may not fully characterize the stability or reproducibility of the observed performance levels. The absence of repeated runs, confidence intervals, or other robustness analyses accordingly limits the strength of the conclusions that can be drawn, particularly for cases where performance differences between configurations are modest. We acknowledge this as a substantive methodological limitation, and note that it was imposed primarily by the computational cost of training multi-stage pipelines at scale on A100 GPU clusters. All reported trends are nonetheless consistent across multiple benchmarks and across qualitatively distinct pipeline comparisons, which provides partial support for their reliability.

Fourth, the coverage of the statistics domain in the training data is inherently incomplete, with certain subfields underrepresented. The limited availability of high-quality statistical chain-of-thought data further constrains both interpretability and generalization. In addition, part of the preference data was distilled from the Gemini API. While this improves data efficiency, it introduces potential risks: (1) teacher-model bias: Gemini’s responses may reflect stylistic conventions, conceptual emphases, or factual framings specific to that model, which could be inadvertently absorbed by StatLLaMA; (2) distributional narrowness: despite prompt-level diversity controls, synthetic corpora may under-represent the breadth and irregularity of real-world statistical reasoning; and (3) evaluation leakage: while our benchmarks (**GSM8K**, **ARC**, **AP Statistics**) were not used as prompts during data generation, we cannot fully rule out distributional overlap between the synthetic training data and benchmark items.

Finally, computational constraints limited the scope of hyperparameter exploration, particularly in the GRPO alignment stage. As a result, we were unable to exhaustively evaluate the full range of GRPO configurations, including LoRA rank, alpha, learning rate, KL coefficient, group size, and data mixing ratios. While the directional trends reported are consistent, additional experimentation could reveal further refinements and might alter some of the specific configuration-level comparisons.

6.3. Future directions

Future work should proceed along several interconnected lines. A primary methodological priority is the introduction of rigorous robustness analyses into the evaluation framework. Specifically, future studies should conduct multiple independent training runs for each key configuration and report standard deviations over benchmark scores. Such analyses are essential for distinguishing reliable performance differences from noise attributable to training stochasticity, and for establishing the reproducibility of the multi-stage training paradigm proposed here.

Closely related is the need to expand the evaluation suite. Future work should incorporate benchmarks that more directly reflect applied statistical analysis and consultation, including tasks that require end-to-end workflow reasoning, interpretation of ambiguous or messy data, and identification of methodological errors. To assess the bias introduced by Gemini-generated synthetic data, it is important to construct a targeted evaluation set of questions known to elicit common teacher-model errors—such as informal definitions of p-values, conflation of statistical and practical significance, or omission of distributional assumptions—and measure StatLLaMA’s error rate against both the Gemini teacher and a human-expert reference. Evaluation on additional held-out benchmarks with minimal distributional overlap with the synthetic training data would also provide stronger evidence for genuine generalization.

Expanding high-quality statistical training datasets, particularly those featuring transparent reasoning traces and human-annotated ground truth, remains a priority. Synthetic data generation via teacher models is a practical and scalable approach, but it introduces risks of stylistic bias, distributional narrowness, and evaluation leakage. We recommend training a variant of StatLLaMA on human-annotated data in place of the Gemini-generated corpus and comparing response characteristics between the two variants, which would provide a direct empirical estimate of teacher-model bias.

Examining the scalability of the proposed pipeline to larger model sizes is also important. It is plausible that at larger scales, base models develop sufficient latent instruction-following capability to make Pipelines 1 or 2 more competitive, or that the trade-offs between domain specialization and general reasoning observed here shift substantially. Controlled scaling experiments would clarify the scope of the current findings.

Methodologically, further study can investigate improved distillation strategies that reduce teacher-model bias while maintaining signal strength, and explore alternative PEFT variants that offer better stability-adaptability trade-offs. A more systematic comparison of DPO and GRPO across model scales, preference dataset sizes, and domain types would also help determine the generality of the alignment conclusions reported here. Methods for structured data augmentation, counterexample generation,

and topic balancing also warrant systematic exploration.

Finally, as statistical LLMs move toward real-world use, issues of explainability, robustness, and uncertainty quantification will become central. Addressing these concerns is essential for responsible deployment in scientific, educational, and decision-support settings.

Overall, this study provides an empirically grounded foundation for building statistically competent, lightweight language models and highlights a practical path forward for domain-specialized LLM development.

Acknowledgments

This research was partially supported by grants from the Ministry of Science and Technology, Taiwan (MOST 111-2118-M-A49-003-MY2) and the National Science and Technology Council, Taiwan (NSTC 113-2118-M-A49-006 and NSTC 114-2118-M-A49-002). We are grateful to the National Center for High-performance Computing, Taiwan for computer time and facilities. During the preparation of this work, the authors used OpenAI's ChatGPT in order to improve language and readability. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

- AP Program (2024). *AP Statistics Course-AP Central*. College Board, <https://apcentral.collegeboard.org/courses/ap-statistics>. Retrieved on April 28, 2024.
- Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., El-Showk, S., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Hume, T., Johnston, S., and et al. (2022). Training a helpful and harmless assistant with reinforcement learning from human feedback. <https://arxiv.org/abs/2204.05862>.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C. et al. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf.
- Calderon, N., Mukherjee, S., Reichart, R., and Kantor, A. (2023). A systematic study of knowledge distillation for natural language generation with pseudo-target training. <https://arxiv.org/abs/2305.02031>.

- Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D. et al. (2024). Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, <http://jmlr.org/papers/v25/23-0870.html>.
- Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. (2018). Think you have solved question answering? Try ARC, the AI2 reasoning challenge. <https://arxiv.org/abs/1803.05457>.
- Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. (2021). Training verifiers to solve math word problems. <https://arxiv.org/abs/2110.14168>.
- Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023a). **bitsandbytes**. <https://github.com/bitsandbytes-foundation/bitsandbytes>.
- Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023b). Qlora: Efficient finetuning of quantized LLMs. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA. Curran Associates Inc. DOI: [10.5555/3666122.3666563](https://doi.org/10.5555/3666122.3666563).
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics, DOI: [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423), <https://aclanthology.org/N19-1423/>.
- Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A. et al. (2024). The llama 3 herd of models. <https://arxiv.org/abs/2407.21783>.
- Gupta, K., Therien, B., Ibrahim, A., Richter, M. L., Anthony, Q., Belilovsky, E., Rish, I., and Lesort, T. (2023). Continual pre-training of large language models: How to (re)warm your model? <https://arxiv.org/abs/2308.04014>.
- Gururangan, S., Marasovic, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. (2020). Don’t stop pretraining: Adapt language models to domains and tasks. <https://arxiv.org/abs/2004.10964>.
- Han, D., Han, M., and Unsloth team (2023). **Unsloth**. <http://github.com/unslothai/unsloth>.
- Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. <https://arxiv.org/abs/1503.02531>.
- Honnibal, M., Montani, I., Van Landeghem, S., and Boyd, A. (2020). **spaCy**: Industrial-strength Natural Language Processing in Python. DOI: [10.5281/zenodo.1212303](https://doi.org/10.5281/zenodo.1212303).

- Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. <https://arxiv.org/abs/1801.06146>.
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). Lora: Low-rank adaptation of large language models. <https://arxiv.org/abs/2106.09685>.
- Huang, J., Cui, L., Wang, A., Yang, C., Liao, X., Song, L., Yao, J., and Su, J. (2024). Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. <https://arxiv.org/abs/2403.01244>.
- Hugging Face (2020). AutoTokenizer: A generic tokenizer class. https://huggingface.co/docs/transformers/v4.56.2/en/model_doc/auto#transformers.AutoTokenizer.
- Ibe, C. (2024). Unlocking low-resource language understanding: Enhancing translation with llama 3 fine-tuning. <https://medium.com/ccibeekeoc42/unlocking-low-resource-language-understanding-enhancing-translation-with-llama-3-fine-tuning-df8f1d04d206>. Accessed 2024-05-30.
- Iyer, V., Rei, R., Chen, P., and Birch, A. (2025). Xl-instruct: Synthetic data for cross-lingual open-ended generation. <https://arxiv.org/abs/2503.22973>.
- Jin, X., Zhang, D., Zhu, H., Xiao, W., Li, S.-W., Wei, X., Arnold, A., and Ren, X. (2022). Lifelong pretraining: Continually adapting language models to emerging corpora. <https://arxiv.org/abs/2110.08534>.
- Jing, L. and Tian, Y. (2021). Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 43(11):4037–4058, ISSN: 1939-3539, DOI: [10.1109/TPAMI.2020.2992393](https://doi.org/10.1109/TPAMI.2020.2992393).
- Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, DOI: [10.1073/pnas.1611835114](https://doi.org/10.1073/pnas.1611835114).
- Lialin, V., Deshpande, V., Yao, X., and Rumshisky, A. (2024). Scaling down to scale up: A guide to parameter-efficient fine-tuning. <https://arxiv.org/abs/2303.15647>.
- Liu, S.-Y., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C. F., Cheng, K.-T., and Chen, M.-H. (2024). Dora: weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning, ICML'24*. JMLR.org. DOI:[10.5555/3692070.3693369](https://doi.org/10.5555/3692070.3693369).

- Longpre, S., Hou, L., Vu, T., Webson, A., Chung, H. W., Tay, Y., Zhou, D., Le, Q. V., Zoph, B., Wei, J., and Roberts, A. (2023). The flan collection: Designing data and methods for effective instruction tuning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22631–22648. PMLR, <https://proceedings.mlr.press/v202/longpre23a.html>.
- Lopez-Paz, D. and Ranzato, M. (2017). Gradient episodic memory for continual learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., https://proceedings.neurips.cc/paper_files/paper/2017/file/f87522788a2be2d171666752f97ddeb-Paper.pdf.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. (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. DOI: [10.5555/3600270.3602281](https://doi.org/10.5555/3600270.3602281).
- Pan, Y., Cai, Z., Chen, G., Zhong, H., and Wang, C. (2025). What matters in data for DPO? <https://arxiv.org/abs/2508.18312>.
- Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. <https://www.mikecaptain.com/resources/pdf/GPT-1.pdf>.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.
- Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. (2023). Direct preference optimization: your language model is secretly a reward model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc. <https://openreview.net/forum?id=HPuSIXJaa9>.
- Rasley, J., Rajbhandari, S., Ruwase, O., and He, Y. (2020). **DeepSpeed**: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pages 350–3506, New York, NY, USA. Association for Computing Machinery, ISBN: [9781450379984](https://doi.org/10.1145/3394486.3406703), DOI: [10.1145/3394486.3406703](https://doi.org/10.1145/3394486.3406703).
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. <https://arxiv.org/abs/1707.06347>.

- Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. (2024). Deepseekmath: Pushing the limits of mathematical reasoning in open language models. <https://arxiv.org/abs/2402.03300>.
- Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. (2023). Alpaca: A strong, replicable instruction-following model. <https://crfm.stanford.edu/2023/03/13/alpaca.html>.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Roziere, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023a). Llama: Open and efficient foundation language models. <https://arxiv.org/abs/2302.13971>.
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B. et al. (2023b). Llama 2: Open foundation and fine-tuned chat models. <https://arxiv.org/abs/2307.09288>.
- Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. (2023). Self-instruct: Aligning language models with self-generated instructions. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *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, DOI: [10.18653/v1/2023.acl-long.754](https://doi.org/10.18653/v1/2023.acl-long.754).
- Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. (2022). Finetuned language models are zero-shot learners. <https://arxiv.org/abs/2109.01652>.
- Zheng, R., Dou, S., Gao, S., Hua, Y., Shen, W., Wang, B., Liu, Y., Jin, S., Liu, Q., Zhou, Y., Xiong, L., Chen, L., Xi, Z., Xu, N., Lai, W., Zhu, M., Chang, C., Yin, Z., Weng, R., Cheng, W., and et al. (2023). Secrets of RLHF in large language models part I: PPO. <https://arxiv.org/abs/2307.04964>.

A. Construction of Training Datasets

To effectively leverage diverse datasets across multiple training stages, this study adopted a systematic dataset construction and integration framework. This process involved not only the cleaning and normalization of raw data (as discussed in Section 4.1) but also the design of data organization methods, mixing strategies, prompt engineering, and format conversions tailored to specific training objectives (CoP, instruction tuning, SFT, and RLHF). These strategies aimed to maximize data utility, ensure effective skill acquisition at each stage, and optimize model learning efficiency under constrained computational resources.

A.1. Continual pretraining

The goal of the CoP stage was to immerse the model in a professional statistical language environment, enabling it to internalize domain-specific terminology, writing conventions, and conceptual knowledge. Two primary datasets were used: academic paper excerpts (**S2ORC**) and statistical terms and definitions (**Statistical Nouns/Defs**). The **S2ORC** subset was precisely filtered using title and subtitle keywords to ensure their relevance to statistical domain. Each selected paper was stored as a dictionary, with the paper title and keywords as the key and paragraph excerpts as values. Structured fields such as “Problem”, “Method”, “Result”, and “Conclusion” were then extracted to form the final dataset. To strengthen understanding of key terminology, the **Statistical Nouns/Defs** data were repeated five times, whereas academic paper excerpts were used once in the training set. The combined dataset was trained for two full epochs.

Apart from the direct use of text corpora in the CoP stage, in the subsequent stages of instruction tuning, SFT, and RLHF, all data were format-converted through a default chat template to train the model’s understanding and generation abilities in conversational interaction scenarios.

A.2. Instruction tuning

The instruction tuning stage established the model’s foundational instruction-following capability. Two large-scale general-purpose instruction-following datasets: **Dolly-15k** and **OpenHermes 2.5** were used, each converted into the chat format [{"role": "user", "content": "..."}, {"role": "assistant", "content": "..."}]. Neither dataset was augmented or repeated; the combined corpus was trained for one epoch. This setup provided broad exposure to general instruction patterns while avoiding overfitting or premature bias toward specific response formats. The primary objective was to quickly build a general instruction-response framework rather than to perform deep optimization on any specific general skill.

A.3. Supervised fine-tuning

The SFT stage enhanced the model’s reasoning, statistical problem-solving, and complex instruction-handling abilities through a carefully constructed mixture of datasets with differentiated augmentation:

- **GSM8K (Train)**: step-by-step mathematical reasoning (no augmentation)
- **Statistical CoT**: chain-of-thought reasoning, repeated 2×
- **Statistical Nouns/Defs**: term definitions, repeated 3×
- **Statistical GRPO**: preferred response style examples used here as supervised signals, repeated 2×

All subsets were merged into a single mixed dataset and trained for three epochs. This differentiated augmentation ensured that more complex or domain-intensive tasks were represented more frequently, allowing the model to develop deeper reasoning and contextual understanding while maintaining balance through **GSM8K**'s unaltered baseline proportion.

In addition, two further datasets: **Math-QA** (mathematical problem QA) and **FineTome-100k** (general-purpose supervised fine-tuning data) were incorporated in variant configurations to examine how different SFT setups influence overall performance.

The **Statistical CoT** and **Statistical GRPO** datasets were generated using Gemini based on their corresponding prompts (Table 6, Appendix C.4). This process introduces potential teacher-model bias, whereby Gemini may produce shallow, imprecise, or conceptually flawed statistical explanations that could be propagated to StatLLaMA. To assess this risk, we conducted a targeted audit of a sample of Gemini-generated **Statistical CoT** responses, reviewed by a domain expert in statistics. The audit did not reveal systematic factual errors.

A.4. Reinforcement learning from human feedback

The RLHF stage aimed to align the model's output with human preference signals through contrastive learning on pairwise preference data. Three datasets were used: **Statistical GRPO**, **Statistical DPO**, and **Math DPO**. In training DPO version of RLHF, **Statistical DPO** and **Math DPO** datasets were unaugmented and combined for two training epochs. By jointly training on preference data from both the mathematics and statistics domains, the model learned preference signals across different task types. This approach enabled it to internalize not only general response quality patterns but also the domain-specific stylistic preferences unique to each field, thereby enhancing its ability to produce precise, human-aligned responses in both mathematical and statistical tasks.

A.5. Summary of dataset construction strategy

In summary, the dataset design and mixing strategies at each training stage were guided by a holistic consideration of data characteristics, training objectives, model learning dynamics, and resource constraints. Repetition factors and epoch counts were systematically adjusted to balance data influence and computational efficiency. This structured approach ensured comprehensive enhancement of the model's performance across all key dimensions—statistical knowledge acquisition, reasoning capability, instruction following, and alignment with human preferences—while maintaining efficiency and stability throughout the training pipeline.

B. Qualitative Evaluation Questions and Outputs

Table 5: Prompts used to evaluate statistical question-answering performance, along with model outputs from StatLLaMA and LLaMA-3.2-3B-Instruct.

Prompt 1:

Explain the difference between descriptive statistics and inferential statistics.

Provide an example of each.

LLaMA-3.2-3B-Instruct:

Descriptive statistics and inferential statistics are two types of statistical methods used to analyze and interpret data. The main difference between them lies in their purpose, scope, and level of inference.

Descriptive Statistics

Descriptive statistics aim to summarize and describe the main features of a dataset. They provide a snapshot of the data, without making assumptions about the population. The main goals are:

1. To summarize the central tendency (mean, median, mode)
2. To summarize variability (range, variance, standard deviation)
3. To describe shape and distribution (skewness, kurtosis)

Example: A researcher describes the average height of 100 students by calculating the mean (175 cm). This summarizes the sample without making population-level inferences.

Inferential Statistics

Inferential statistics aim to make inferences about a population based on a sample. The main goals are:

1. To estimate population parameters (mean, proportion)
2. To test hypotheses (e.g., is the population mean equal to a known value?)
3. To make predictions about the population

Example: A researcher collects a sample of 100 adults to test if the population mean height exceeds 175 cm. If the sample mean is greater, the researcher infers that the population mean is likely greater as well.

Summary

Descriptive statistics summarize data; inferential statistics draw conclusions about populations from samples.

StatLLaMA:

Descriptive statistics and inferential statistics are two major types of methods used to analyze data.

Descriptive Statistics

Descriptive statistics summarize and describe the basic features of a dataset, helping to understand characteristics such as central tendency, variability, and distribution. They provide a snapshot of the data without inferring population properties.

Example: Given exam scores from 100 students, we can compute the mean, median, mode, and range to describe the class’s performance distribution.

Inferential Statistics

Inferential statistics are used to make generalizations about a population based on a sample. They allow conclusions about characteristics, relationships, or behaviors of the population.

Example: To determine whether the average exam score in one school is higher than the national average, we compute a confidence interval for the sample mean. If this interval exceeds the national average, we infer the school’s average is likely higher.

Key Differences

- Descriptive statistics summarize observed data, while inferential statistics make predictions about the population.
- Descriptive methods describe characteristics; inferential methods generalize findings.
- Descriptive analysis does not involve inference, whereas inferential analysis uses probability to draw conclusions.

Table 5 – Continued from previous page

Prompt 2:

What are some key properties or assumptions associated with “Linear Regression Model” as defined in the knowledge graph? Based on triples like: (Linear Regression Model, has_assumption, linearity), (Linear Regression Model, has_assumption, independence of errors), (Linear Regression Model, has_assumption, homoscedasticity), (Linear Regression Model, has_assumption, normality of errors)

LLaMA-3.2-3B-Instruct:

Based on the provided triples, here are some key properties or assumptions associated with a Linear Regression Model:

1. **Linearity:** The relationship between the independent variables and the dependent variable is assumed to be linear.
2. **Independence of Errors:** The errors (residuals) are assumed to be independent of each other.
3. **Homoscedasticity:** The variance of the errors is assumed to be constant across all levels of the independent variables.
4. **Normality of Errors:** The errors are assumed to be normally distributed.
5. **Additivity:** The relationship between the independent variables and the dependent variable is assumed to be additive, i.e., the effect of one independent variable is independent of the effects of other independent variables.
6. **No multicollinearity:** The independent variables are assumed to be uncorrelated with each other.
7. **No omitted variables:** The model assumes that all relevant variables are included in the model.

These assumptions are essential for the validity and reliability of linear regression models. However, it’s worth noting that in practice, it’s often difficult to meet all these assumptions simultaneously, and linear regression models may need to be adjusted or modified to account for any violations.

StatLLaMA:

Based on the knowledge graph, the key properties or assumptions associated with a Linear Regression Model are:

1. **Linearity:** The relationship between the independent variables and the dependent variable is assumed to be linear.
2. **Independence of Errors:** The errors in the model are assumed to be independent of each other.
3. **Homoscedasticity:** The variance of the errors is assumed to be constant across all levels of the independent variables.
4. **Normality of Errors:** The errors are assumed to be normally distributed.

These assumptions are fundamental to the Linear Regression Model and are used to estimate the coefficients of the model. Violations of these assumptions can lead to biased or inconsistent estimates, and may require additional steps to account for the violations.

Continued on next page

Table 5 – Continued from previous page

Prompt 3:

Hi, I'm a medical intern analyzing patient blood pressure data collected across different times of day. My supervisor asked me to check whether time of measurement significantly affects blood pressure. I thought of using ANOVA, but I'm confused because the same patients appear in multiple time slots. Should I be using repeated measures ANOVA instead?

LLaMA-3.2-3B-Instruct:

You're correct to consider the repeated measures aspect of your data. Since the same patients appear in multiple time slots, you're dealing with a type of repeated measures data.

ANOVA (Analysis of Variance) is typically used when you have independent groups being compared to each other, whereas in your case, you have multiple measurements from the same group (patients) at different time points.

Repeated Measures ANOVA (RM-ANOVA) is the appropriate statistical test to use in your situation. RM-ANOVA compares the means of multiple measurements from the same group to detect significant differences between the measurements. Here's a brief summary of the key differences:

- **ANOVA:** Compares means of independent groups (e.g., different patients, treatments, or conditions) to detect significant differences between the groups.
- **Repeated Measures ANOVA (RM-ANOVA):** Compares means of multiple measurements from the same group (e.g., different time points, patients, or conditions) to detect significant differences between the measurements.

When using RM-ANOVA, you'll need to consider the following:

- **Repeatability:** You need to ensure that the measurements are repeatable, meaning that the same patient's blood pressure is measured at the same time on multiple occasions.
- **Randomization:** You should randomize the order of measurements to minimize any biases.
- **Outliers:** You'll need to handle any outliers that may affect the results.

By using RM-ANOVA, you'll be able to determine whether the time of measurement significantly affects blood pressure in your patients.

It's always a good idea to consult with your supervisor or a statistician for further guidance on your specific data and research question.

StatLLaMA:

In your case, since the same patients appear in multiple time slots, you're dealing with a repeated measures design. This means that you're essentially measuring the same variable (blood pressure) at different times (time slots) for the same subjects.

Using repeated measures ANOVA is the correct approach here. Here's why:

1. **Subject variability:** Since the same patients are measured multiple times, subject variability is a concern. Repeated measures ANOVA accounts for this variability by including it as a random effect in the model.
2. **Dependent variable:** Blood pressure is a continuous dependent variable, which is suitable for repeated measures ANOVA.
3. **Time slots:** You have multiple time slots, which are the levels of the repeated measure. Repeated measures ANOVA allows you to compare the means of blood pressure across these time slots.

ANOVA, on the other hand, is typically used for independent samples (e.g., comparing two groups) and doesn't account for subject variability.

When using repeated measures ANOVA, keep in mind:

1. **Assumptions:** Check the assumptions of repeated measures ANOVA, such as normality of the data and equal variances across time slots.
2. **Model complexity:** Consider whether additional models, such as mixed-effects models or generalized linear mixed models, might be more suitable for your data.

In your case, using repeated measures ANOVA should help you determine whether time of measurement significantly affects blood pressure.

C. Data Preprocessing

Transforming raw, heterogeneous data into high-quality input suitable for LLM training is a foundational step in building a domain-specific model. This process demands not only precise technical implementation but also a careful understanding of diverse data characteristics, potential pitfalls, and effective processing strategies. The following subsections outline the specific practices and technical considerations applied in this study to prepare heterogeneous statistics-related data for training and alignment.

C.1. Basic general processing

The first stage of all preprocessing workflows involves cleaning and standardizing raw text. The goal is to establish a clean, consistent foundation for downstream processing and training. Using standard regular expressions and string operations, we removed residual web artifacts (e.g., HTML/XML tags) and normalized whitespace, newlines, and tab characters. Text encodings were converted to UTF-8 to avoid character-set inconsistencies.

To address structural noise from web-crawled sources (such as repeated headers, footers, and page numbers), we designed a pattern recognition mechanism for automatic filtering. For academic texts, reference sections were removed using keyword detection (e.g., “References”, “Bibliography”) to increase corpus knowledge density. Finally, while narrative text was lowercased for consistency, statistical terms and mathematical symbols were preserved in their original form.

C.2. Extracting structured knowledge from text

As a supplementary exploration, we attempted to extract structured semantic knowledge directly from unstructured academic text. Using **SpaCy**'s `en_core_web_lg` model (Honnibal et al. 2020) with dependency parsing, we extracted subject-verb-object (SVO) triplets. In principle, this approach could support automatic construction of a semantic network. However, the complexity of statistical writing—long sentences, subordinate clauses, and technical terminology—led to low accuracy and limited coverage compared to curated ontologies. As such, these results were excluded from the main training corpus and retained only for auxiliary analysis.

C.3. Unified formatting and tokenization

Once cleaning and structural modeling were complete, all text was converted into numerical sequences for training. We adopted a pre-tokenization strategy, ensuring all text was tokenized and ID-mapped prior to training, which reduced overhead and improved efficiency. Tokenization was performed using Hugging Face's AutoTokenizer (Hugging Face 2020), configured to match the vocabulary and algorithm of the base model. Texts exceeding the model's maximum sequence length were processed using a sliding window approach to preserve context, while shorter inputs were padded. Padding tokens were assigned a label value of -100 , ensuring they did not affect loss computation.

C.4. Prompt engineering

Prompt engineering played a central role throughout data preparation, training, and evaluation. The design quality of prompts directly influenced model learning efficiency and downstream performance. To address different tasks and training stages, we developed multiple prompt templates, including:

- Prompts for creating CoT QA pairs in statistics.
- Prompts for producing preference samples (question, reasoning, answer) for GRPO and (prompt–chosen–rejected) for DPO training.
- Benchmark-specific evaluation prompts to ensure fairness and comparability.

All prompt templates were applied after tokenization, converting the data into structured inputs with standardized response formats.

Through iterative refinement, carefully designed prompts substantially improved the model’s ability to generate reliable, task-specific outputs, thereby enhancing both performance and applicability in statistics. Several design choices were intended to reduce teacher-model bias from the outset. The Statistical CoT prompts explicitly required step-by-step reasoning, coverage of specific statistical subfields, and pedagogical rigor, eliciting formally grounded responses rather than conversational ones. The Statistical DPO construction required “chosen” responses to be conceptually accurate and well-structured while “rejected” responses contained identifiable errors, meaning StatLLaMA was trained to prefer formal correctness over stylistic fluency.

The full set of prompt templates is provided in Table 6.

Table 6: Full set of prompt templates used in data preparation. Each dataset is generated by Gemini using its corresponding prompt, encompassing diverse statistical reasoning and dialogue tasks to ensure comprehensive understanding across training strategies.

Dataset	Prompt
Statistical CoT	<p>You are a distinguished Statistics Professor and an expert in educational design. Your task is to generate {num_to_generate} high-quality statistical problem sets. Each problem must include a clear, step-by-step <i>chain-of-thought</i> reasoning process. Accuracy, clarity, and pedagogical rigor are essential.</p> <p>Instructions:</p> <ol style="list-style-type: none"> Output Format: Produce a single JSON list containing {num_to_generate} dictionary objects. Do not include markdown indicators such as "json". Each object must contain the keys "question" and "cot". Example: {"question": "...", "cot": ["1. ...", "2. ...", "3. ..."]} Content Guidelines: - "question": A clear, multi-step statistical question covering topics such as probability theory (Bayes' theorem, conditional probability), distributions (Binomial, Poisson, Normal), sampling, confidence intervals, hypothesis testing (t-tests, chi-squared tests), correlation, and regression. - "cot": A list of sequential reasoning steps written as strings. Each step should progress logically toward the final answer. All formulas and variables must be written in valid LaTeX notation.
Statistical GRPO	<p>You are a distinguished Statistics Professor known for creating clear and instructive problems. Your task is to generate {num_to_generate} high-quality Question-Answer (QA) sets for Group Relative Policy Optimization (GRPO).</p> <p>Instructions:</p> <ol style="list-style-type: none"> Output Format: Create a single JSON list of dictionaries with keys "question", "reasoning", and "answer". All numerical values must be represented as strings. Example: {"question": "...", "reasoning": "...", "answer": "0.375"}. Content Requirements: - "question": Pose a clear statistical question requiring analytical reasoning and multi-step problem solving. Cover diverse areas such as descriptive statistics, probability, distributions, sampling, confidence intervals, hypothesis testing, correlation, and regression. Include both conceptual and computational tasks, avoiding overly simplistic textbook examples. - "reasoning": Provide a detailed, accurate explanation with all formulas in LaTeX notation. - "answer": Present the definitive answer derived from the reasoning. Ensure consistency with any format requirements specified in the question.
Statistical DPO	<p>You are an AI assistant tasked with generating training data for Direct Preference Optimization (DPO). Produce {num_to_generate} distinct data points encompassing various statistical concepts within a single response.</p> <p>Instructions:</p> <ol style="list-style-type: none"> Output Format: Generate one valid JSON list where each element contains the keys "prompt", "chosen", and "rejected". Example: [{"prompt": "...", "chosen": "...", "rejected": "..."}, ...]. Content Guidelines: - Topic Diversity: Cover different statistical domains, including descriptive statistics, probability, inference, regression, and study design. Aim for a mix of difficulty levels. - "prompt": Vary prompt styles — conceptual explanations, comparisons, applications, or reasoning-based tasks. - "chosen": Provide accurate, clear, and well-structured answers with sound logic. Use good examples/ analogies where appropriate. - "rejected": Address the same prompt but include subtle inaccuracies, misconceptions, or unclear reasoning to create a plausible yet inferior response. It should not be completely wrong or nonsensical.

D. Ablation of CoP in Pipelines 1 and 2

To assess the contribution of CoP in Pipelines 1 and 2, we conducted targeted ablation experiments in which the CoP stage was removed. The results are reported in Table 7.

Table 7: Benchmark results of removing the CoP stage from Pipelines 1 and 2.

Model	GSM8K (Test)	AP Statistics	ARC
	(8-shot + CoT, Acc. %)	(0-shot, Acc. %)	(0-shot, Acc. %)
LLaMA-3.2-3B	29.87	26.13	42.24
+ SFT	34.27	26.13	43.53
+ SFT + RLHF	28.20	26.83	43.36
+ Instruct	26.91	25.78	42.09
+ Instruct + SFT	36.85	26.13	43.53
+ Instruct + SFT + RLHF	27.07	26.13	43.19

Values in boldface denote the best result for each benchmark.

Eliminating CoP yields nearly identical performance across all benchmarks, indicating that it provides little to no measurable benefit in this setting. A plausible explanation is that the CoP datasets (`S2ORC` and `Statistical Nouns/Defs`) substantially overlap with the original pretraining corpus of LLaMA-3.2-3B, resulting in minimal information gain.

E. Designs and Benchmark Results for Pipeline 3

Pipeline 3 constitutes the core experimental direction of this study. Its central strategy is to directly leverage the strong general language understanding and instruction-following capabilities of LLaMA-3.2-3B-Instruct, using it as an efficient starting point for specialization in the statistics domain. Within this pipeline, we designed and rigorously compared three main SFT strategies (v1, v2, and v3.x) to systematically evaluate the combined effects of different fine-tuning methods, data mixing schemes, and preference optimization techniques (GRPO vs. DPO).

E.1. SFT stage

In the SFT stage of Pipeline 3, our primary goal was to steer the general capabilities of the LLaMA-3.2-3B-Instruct model toward statistics-focused reasoning and expression. To that end, we implemented three distinct SFT strategies, each reflecting a different philosophy of training and data integration.

SFT-v1 (Phased, differentiated strategy). The v1 strategy adopted a multi-phase, template-specific design that emphasized hierarchical capability building:

- Phase 1: Strengthened mathematical and statistical reasoning through instruction-style SFT on chain-of-thought (`Statistical CoT`) and mathematical problem QA (`Math-QA`).

- Phase 2: Expanded domain expression using S2ORC-derived summary/method/conclusion segments, fine-tuned in an instruction style to enhance statistical narrative fluency.
- Phase 3: Applied a chat template to separately fine-tune on `FineTome-100k` (general conversation) and `Statistical Nouns/Defs` (terminology). LoRA adapter weights from both fine-tunings were then merged via linear interpolation (weighted averaging for the same key values), aiming to integrate domain knowledge, term understanding, and general interaction ability into a single adapter.

SFT-v2 (Unified, single-stage strategy). Reflecting on the complexity and fragility of v1, the v2 strategy pursued a simpler, unified training approach. From the outset, all data were formatted with a chat template, and the following datasets were combined into a single mixed training set: `Statistical Nouns/Defs`, `Statistical CoT`, `Statistical GRPO` (preference-style data used here as supervised signals), `FineTome-100k`, and `GSM8K (Train)`. This mixture was used for a one-stage SFT with a higher LoRA rank (rank = 32). The goal was to cultivate comprehensive statistical reasoning and conversational ability within a consistent interaction framework, thereby providing a solid base for subsequent preference optimization.

SFT-v3.x (Controlled variants based on v2). The v3.x series built upon the unified template design of v2 but introduced controlled variations to examine the effect of different data mixtures and hyperparameters (Table 8).

Table 8: Configuration of data mixtures and hyperparameters for the SFT-v3.x series of Pipeline 3.

Version	Statistical Nouns/Defs ¹	Statistical CoT ¹	Statistical GRPO ¹	FineTome -100k ¹	GSM8K (Train) ¹	train_on _responses_only ²	Epoch ³
SFT-v3.1	3 times	1 time	1 time	1 time	1 time	○	3
SFT-v3.2	3 times	2 times	2 times	1 time	1 time	○	3
SFT-v3.3	3 times	1 time	1 time	1 time	1 time	×	3
SFT-v3.4	3 times	2 times	2 times	×	1 time	○	3
SFT-v3.5	×	×	×	×	1 time	○	1
SFT-v3.6	3 times	1 time	1 time	×	×	○	1

¹ The entries “1 time”, “2 times”, and “3 times” indicate the repetition factor for each source dataset when constructing the final mixed training set. A × symbol in a dataset column indicates that the dataset was not used in that version.

² `train_on_responses_only`: ○ indicates loss computation only on the response; × indicates loss on the full sequence.

³ Number of training epochs.

The benchmark results for all SFT variants of Pipeline 3 are reported in Table 9. From Table 9, several conclusions emerge:

1. Domain gains on `AP Statistics`. All SFT variants improve `AP Statistics` relative to the original `Instruct` baseline (37.63% up to 41.46%), confirming that SFT is effective at injecting domain-specific statistical knowledge.
2. Trade-offs across benchmarks. SFT-v1 shows modest `AP Statistics` gains and relatively stable `ARC`, but suffers a notable drop in `GSM8K (Test)`, suggesting that

its multi-stage complexity does not translate into superior domain performance. SFT-v2 improves **AP Statistics** more clearly and maintains **GSM8K (Test)** and **ARC** at levels comparable to v1, providing a more balanced profile. The v3.x series reveals a nuanced trade-off surface: Configurations like v3.1, v3.3, and v3.4 push **AP Statistics** higher but tend to reduce **GSM8K (Test)** and, in some cases, **ARC** (e.g., v3.3). v3.6 better preserves **GSM8K (Test)** and **ARC** but sacrifices some **AP Statistics** gains.

3. Impact of `train_on_responses_only` and data composition. Differences between v3.2 vs. v3.3 or v3.4 vs. v3.6 highlight that both the loss computation setting and dataset composition/repetition play important roles in shaping the balance between general and domain-specific abilities.

In summary, the SFT exploration underscores that no single strategy is universally optimal; the best choice depends on the desired trade-off between general reasoning and statistical expertise. SFT-v2 offers a strong, balanced baseline, while variants like v3.3 and v3.4 provide stronger statistical performance at some cost to general abilities. These models (especially v2, v3.3, and v3.4) were selected as candidates for the subsequent preference optimization stage.

Table 9: Benchmark results of SFT strategies (v1, v2, and v3.x series) in Pipeline 3.

Model	GSM8K (Test) (8-shot + CoT, Acc. %)	AP Statistics (0-shot, Acc. %)	ARC (0-shot, Acc. %)
LLaMA-3.2-3B-Instruct	64.44	37.63	43.60
+ SFT-v1	57.92	37.98	41.89
+ SFT-v2	58.53	39.72	40.36
+ SFT-v3.1	51.18	41.11	40.10
+ SFT-v3.2	50.80	40.77	40.61
+ SFT-v3.3	53.60	41.46	36.77
+ SFT-v3.4	54.59	41.11	40.61
+ SFT-v3.5	55.08	39.72	39.85
+ SFT-v3.6	58.15	38.68	41.98

Values in boldface denote the best result for each benchmark.

E.2. RLHF preference optimization: GRPO experiments

After obtaining a set of SFT models with different capability profiles, we moved to the RLHF preference optimization stage, aiming to use preference-based supervision to improve output quality, coherence, and alignment with human expectations beyond what standard SFT can achieve.

As an initial attempt along the RLHF-style preference optimization path, we applied group relative policy optimization (GRPO) to the SFT-v1, SFT-v2, and SFT-v3.4 models. For SFT-v2, we further explored three GRPO configurations by varying: LoRA rank and alpha, and the data mixing ratio between **Statistical GRPO** and **GSM8K (Train)** (mathematics domain) samples. Table 10 summarizes the GRPO settings, and Table 11 reports the benchmark scores.

Table 10: GRPO hyperparameter configurations used with SFT-v2 in Pipeline 3.

Version	LoRA Rank & Alpha	Data Mixing Ratio ¹ (Statistical GRPO : GSM8K (Train))
GRPO-v1	32	2255 : 4000
GRPO-v2	8	2255 : 1000
GRPO-v3	16	2255 : 1000
GRPO-v4	32	2255 : 1000

¹ The sample ratio between Statistical GRPO data and GSM8K (Train) data used for training.

Table 11: Benchmark results from the GRPO preference optimization stage used with SFT-v1, SFT-v2, and SFT-v3.4 in Pipeline 3.

Model	GSM8K (Test) (8-shot + CoT, Acc. %)	AP Statistics (0-shot, Acc. %)	ARC (0-shot, Acc. %)
LLaMA-3.2-3B-Instruct	64.44	37.63	43.60
+ SFT-v1	57.92	37.98	41.89
+ SFT-v1 + GRPO-v2	58.45	38.33	42.32
+ SFT-v2	58.53	39.72	40.36
+ SFT-v2 + GRPO-v1	55.42	40.07	39.85
+ SFT-v2 + GRPO-v2	53.98	40.42	39.85
+ SFT-v2 + GRPO-v3	57.16	40.07	35.32
+ SFT-v2 + GRPO-v4	54.81	40.07	35.06
+ SFT-v3.4	54.59	41.11	40.61
+ SFT-v3.4 + GRPO-v2	56.48	41.46	41.13

Values in boldface denote the best result for each benchmark.

The GRPO results reveal several issues. For SFT-v1, GRPO-v2 yields only minor gains in all three benchmarks, suggesting limited domain enhancement. For SFT-v2, the behavior is strongly dependent on hyperparameters: GRPO-v1 (highest rank, alpha, and largest GSM8K (Train) share) slightly increases AP Statistics but significantly reduces GSM8K (Test). GRPO-v2 (lowest rank, alpha, and less GSM8K (Train)) improves AP Statistics but at the cost of a substantial drop in GSM8K (Test). GRPO-v3 (intermediate settings) partially recovers GSM8K (Test) but fails to surpass GRPO-v2 on AP Statistics and causes a severe degradation on ARC, approaching a catastrophic drop. GRPO-v4, which further increases the rank relative to GRPO-v3, degrades performance across all three benchmarks. A similar pattern is observed for SFT-v3.4, where GRPO-v2 again yields limited and imbalanced gains.

The structured analysis across GRPO configurations illustrates that increasing the GSM8K training share in the data mixture (GRPO-v1) partially preserved mathematical reasoning but suppressed statistical domain gains, while reducing it (GRPO-v2) reversed this trade-off. Lowering LoRA rank reduced the magnitude of instability somewhat but did not eliminate cross-benchmark redistribution. The intermediate configurations (GRPO-v3 and GRPO-v4) achieved the worst outcome on ARC, suggesting

that partial mitigation of one instability dimension can inadvertently destabilize a third benchmark. Taken together, these results indicate that GRPO’s sensitivity reflects a systemic interaction between group-relative optimization dynamics and the structural properties of narrow-domain preference data, rather than any single misconfigured hyperparameter.

Overall, GRPO proved highly sensitive to hyperparameter choices and data mixing ratios, often redistributing performance rather than reliably improving it. In our setting, it did not provide stable, consistent, or controllable gains and frequently harmed general abilities in exchange for modest domain improvements. Given its instability and complexity, we concluded that GRPO was not a suitable primary alignment method for this study.

Affiliation:

Jing-Yi Zeng
Institute of Statistics
National Yang Ming Chiao Tung University
No. 1001, Daxue Rd., East Dist.
Hsinchu City 300093, Taiwan
E-mail: a0987395527@gmail.com

Guan-Hua Huang
Institute of Statistics
National Yang Ming Chiao Tung University
No. 1001, Daxue Rd., East Dist.
Hsinchu City 300093, Taiwan
E-mail: ghuang@nycu.edu.tw
URL: <https://ghuang.stat.nycu.edu.tw/>