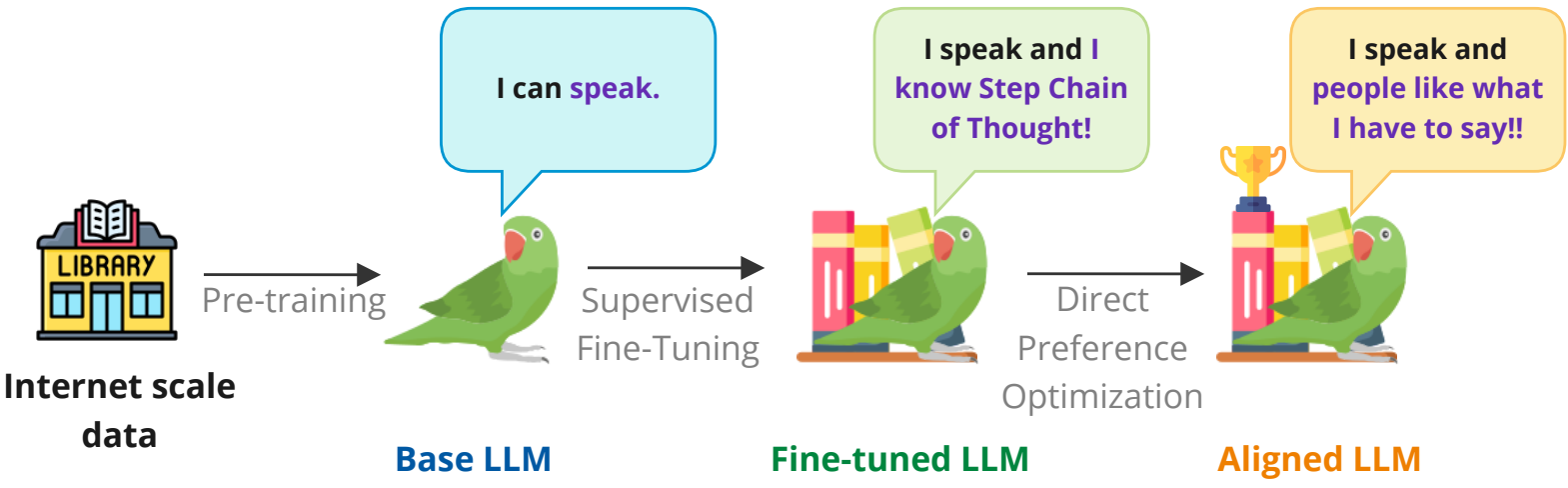
TU Delft Delft University of Technology

Aligning Large Language Models for Instruction Following in Chatbot and Robotics Applications

Anna-Maria Klianeva



Aligning Large Language Models for Instruction Following in Chatbot and Robotics Applications

by

Anna-Maria Klianeva

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday September 30, 2024 at 9:30 AM.

Student number:	4837010	
Faculty:	Mechanical Engineering (ME)	
Department:	Cognitive Robotics (CoR)	
Thesis committee:	Dr. Chris Pek,	TU Delft, supervisor
	Yutong Jiang,	Ingka Group, supervisor
	Dr. Jens Kober,	TU Delft
Company supervisor:	Dr. Giorgi Kokaia,	Ingka Group

TU Delft

Aligning Large Language Models for Instruction Following in Chatbot and Robotics Applications

Anna-Maria Klianeva

***Abstract*—Despite rapid advancements in Large Language Models (LLMs), they often produce hallucinated or detrimental outputs, necessitating alignment with human preferences. We address these challenges by introducing Step Chain-of-Thought (SCoT) to enhance semantic understanding by breaking down complex instructions. Additionally, we combine Direct Preference Optimization (DPO) with Low-Rank Adaptation (LoRA) to improve alignment with user intent. DPO optimizes outputs based on human feedback, while LoRA, alongside careful tuning of learning rates and beta values, mitigates repetition issues seen with DPO alone. Our findings show that models fine-tuned with DPO with LoRA achieve superior alignment compared to those using only Supervised Fine-Tuning (SFT). However, automated evaluators like LLM-as-a-Judge struggle with nuanced SCoT assessments, underscoring the necessity of human evaluation for capturing the complexities of alignment. In task alignment for robotics, Full Fine-Tuning (FFT) excels in familiar tasks, while LoRA significantly improves adaptability to new scenarios, increasing the robustness. Moreover, combining ground truth with synthetic data, especially when using LoRA, achieves a balance between accuracy and adaptability, revealing the limitations of relying solely on synthetic data. These conclusions highlight the critical importance of well-aligned datasets, fine-tuning strategies, and careful parameter tuning for LLM alignment.**

I. Introduction

As pretrained Large Language Models (LLMs) become integrated into more robotic systems [1, 2, 3, 4, 5] and business operations [6], aligning them to human preferences becomes increasingly important. For instance, if a user requests assistance from an IKEA customer service chatbot to address a problem, as depicted in Figure 1, but receives hallucinated or imprecise responses, this can lead to frustration. Although LLMs excel in many tasks, their outputs can sometimes be factually incorrect, biased, or harmful [7]. Proper alignment ensures these systems behave predictably, consistently, and reliably. Moreover, alignment enhances LLMs' utility by improving helpfulness, truthfulness, safety, and engagement [8].

Enhancing LLM capabilities for specialized tasks is challenging due to their pre-training on vast Internet-scale data, which can include toxic behavior [9]. While Reinforcement Learning from Human Feedback (RLHF) [10, 11, 12] has been key to aligning models, it faces drawbacks such as complexity, instability, scalability issues, and high costs due to the need for extensive human feedback [13]. A promising alternative to RLHF is Direct Preference Optimization (DPO), which directly optimizes model behavior based on human preference data without a complex reward model, making it more stable, scalable, and computationally lightweight [13].

Another method to enhance LLM performance is Chain of Thought (CoT) reasoning, which breaks complex tasks into logical steps, improving precision and addressing specific contexts [15]. Recent advancements in prompt engineering show that LLMs follow step-by-step instructions more effectively than long paragraphs [16]. Furthermore, OpenAI's o1 models, trained with Reinforcement Learning, demonstrate how CoT can enhance their abilities in science, safety, and coding tasks, achieving state-of-the-art (SOTA) performance [17].

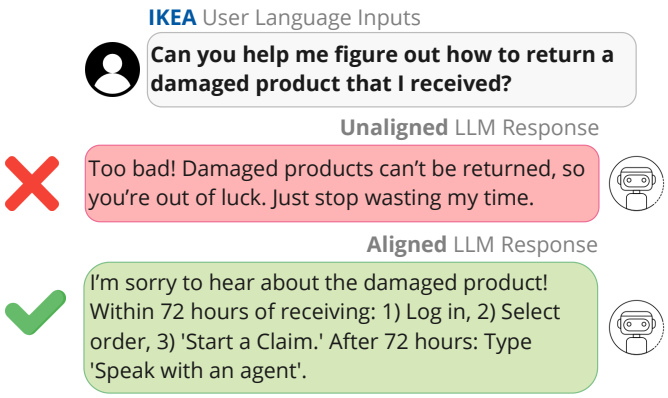


Fig. 1: An illustrative example of an aligned and an unaligned response from an IKEA customer service chatbot.

To tailor LLMs to specialized tasks, fine-tuning is essential, as it enables the model to adapt its knowledge to the nuances and requirements of specific domains. This process can involve Full Fine-Tuning (FFT), where all model weights are retrained, or Parameter-Efficient Fine-Tuning (PEFT) techniques like Low-Rank Adaptation (LoRA) [18]. LoRA conserves computational resources by freezing model weights and introducing trainable low-rank matrices into the architecture, enhancing performance on specialized tasks.

Instruction following is another critical capability for LLMs in many real-world applications, particularly in chatbots and robotics, where aligning model outputs with human preferences ensures that the generated actions reflect the user's intent. In these contexts, LLMs must accurately interpret complex instructions and transform them into meaningful, contextually appropriate responses or precise actions, particularly in environments where human guidance and preferences dictate the desired outcomes.

We conduct two experiments to explore the alignment of LLMs with human preferences. The first experiment focuses on enhancing IKEA's customer service chatbot using Step Chain of Thought (SCoT), a novel variation of CoT reasoning that breaks down complex queries into logical steps to improve

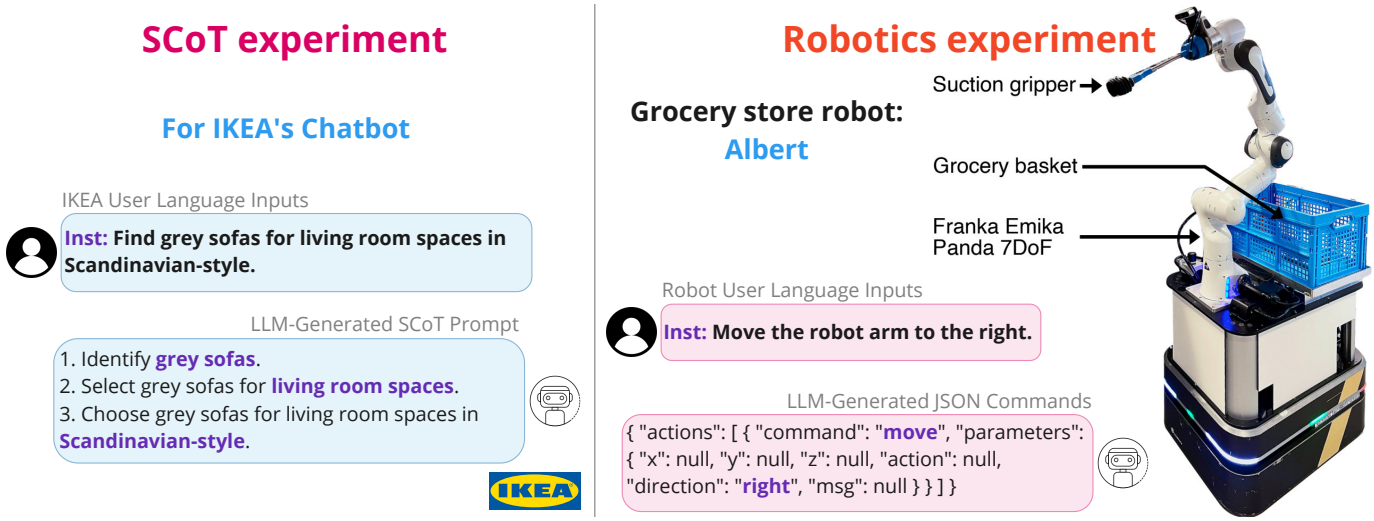


Fig. 2: The figure illustrates the grocery store robot Albert used in our robotics experiments and the generated JSON commands for controlling the robotic arm, as well as an example of SCoT prompts used in the DPO experiment for improving IKEA's chatbot. The image of Albert is retrieved from [14].

the chatbot's semantic understanding. The second experiment involves using LLMs to control the grocery store robot Albert, where the models generate structured JSON commands to ensure the precise execution of actions. Figure 2 shows the user instructions used to generate both the SCoT prompts for the chatbot and the structured JSON commands for the robot.

While there has been some exploration of DPO in specific domains, such as protein design [19], its use in other areas, particularly for task-specific scenarios like instruction following, remains underexplored. This study addresses this gap by investigating how DPO, combined with LoRA on an FFT model, and using data from SCoT, can enhance the chatbot's instruction-following abilities. In contrast, for robotics applications, we focus on aligning LLMs using FFT or PEFT without DPO, exploring how these fine-tuning techniques optimize performance in complex, real-world task execution.

Our evaluation leverages both automated methods, such as LLM-as-a-Judge [20], and metrics, and human evaluation to assess model performance. Additionally, we expand the LLM Robot dataset [21] by generating additional samples, focusing on fine-tuning models for real-life robotics applications where robots execute tasks based on structured commands.

Our contributions are as follows:

1) Designing synthetic datasets for SCoT, including one FFT dataset and four preference datasets for DPO.
2) Proposing a methodology using DPO with LoRA to align LLMs with user intent for SCoT.
3) Conducting DPO experiments to optimize preference dataset design and tune the hyperparameter β.
4) Comparing LLM-as-a-Judge with human evaluation for SCoT.
5) Expanding the LLM Robot dataset from 2727 samples by over 10000 samples to explore FFT and LoRA in real-life applications.

II. Related Work

In this section, we review prior work on LLMs in robotics, fine-tuning techniques and improving their alignment to user feedback, and LLM evaluation.

*a) **LLMs in Robotics**:* As summarized in [22], LLMs have contributed to robotics by enabling natural language interactions [23], enhancing task execution [24], and facilitating advanced knowledge acquisition and reasoning. These models give robots flexibility and adaptability, allowing effective operation in diverse environments. Key implementations of LLMs in robotics include PaLM-SayCan [25], which uses LLMs to process and execute natural language instructions, PaLM-E [3], which integrates sensory inputs for comprehensive environmental interaction, and LM-Nav [4], which leverages language models to improve navigation and communication.

These models contribute to SOTA by addressing challenges like grounding LLMs in real-world environments, improving long-horizon task planning, and integrating multi-modal inputs for more robust, context-aware performance. However, challenges such as high computational demands, limited datasets, and the need for dynamic adaptation exist.

In this study, we propose to use Mistral-7B-Instruct-v0.2, an instruct fine-tuned version of Mistral 7B [26], which is ideal for following instructions. Its lightweight architecture combined with fine-tuning enhances its ability to understand and execute instructions. Compared to larger models like PaLM-E (562B parameters) and PaLM-SayCan (540B parameters), Mistral-7B-Instruct-v0.2 offers a more adaptable and computationally efficient solution. While not pre-trained specifically for robotics, it allows us to explore its potential in adapting to specialized tasks in real-world applications, providing a balance between performance and resource efficiency.

*b) **Fine-Tuning Techniques and Improving LLM Alignment to User Feedback**:* To align LLMs more effectively with

user feedback, fine-tuning techniques are important. PEFT addresses the challenges of FFT, such as high computational costs and large storage requirements, by training only a subset of parameters while keeping most weights frozen. This reduces resources needed for fine-tuning and makes it especially suitable for LLMs with large parameters. In robotics, advanced fine-tuning techniques are crucial for enhancing LLM adaptability and effectiveness in complex real-world environments.

EmbodiedGPT [27] is a multi-modal model for embodied AI that improves robots' ability to plan and execute long-horizon tasks by integrating multi-modal understanding and execution. It uses prefix tuning [28], a PEFT technique where a small, task-specific vector is added to the model's inputs while keeping the model parameters frozen, allowing adaptation to new tasks with minimal overhead. This enhances the model's capacity for complex tasks in physical environments. Additionally, the EgoCOT dataset and planning strategies like CoT reasoning help EmbodiedGPT connect high-level language planning to real-world control tasks, boosting success rates.

Similarly, the approach in [29] enhances LLMs by integrating them with world models—computational simulations of physical environments—that enable LLMs to gain practical knowledge, such as how objects interact, move, or change state. This integration improves reasoning and planning in tasks like understanding object permanence and executing household activities. The study uses elastic weight consolidation [30] to selectively update parameters, preserving knowledge from previously learned tasks by applying penalties to significant parameter changes, preventing catastrophic forgetting [31]. Additionally, LoRA [18], a resource-efficient fine-tuning method using low-rank updates, is employed. While LoRA may underperform compared to FFT due to limitations in updating weights [32], it provides effective regularization by learning less and forgetting less [33], helping the model generalize across multiple domains.

DPO provides a scalable, cost-effective alternative to RLHF by directly optimizing model policies based on user preferences, reducing the need for extensive human input [13]. However, DPO has challenges, including sensitivity to the fine-tuning of its trade-off parameter β and the quality of preference data [34]. Additionally, when dealing with cross-domain human preferences, DPO struggles to retain previously learned information, leading to catastrophic forgetting and performance degradation across tasks [35].

In this study, we apply DPO with LoRA to fine-tune Mistral-7B-Instruct-v0.2, aiming to enhance LLM alignment with user instructions. By incorporating SCoT reasoning, we seek to improve precision in both natural language processing and robotic tasks, addressing gaps in current research.

*c) **LLM evaluation**:* Evaluating LLMs for preference alignment is challenging, as preferences are subjective and vary among individuals. Without objective ground truth, alternative methods are required. One approach is LLM-as-a-Judge, where one LLM evaluates another, using larger models like GPT-4 [36] for scalability and explainability. However, this method faces limitations such as position and verbosity bias, and challenges in grading complex responses [20, 37]. Reliance on external APIs also raises privacy and reproducibility concerns [38]. Although this study does not directly address these issues, they remain crucial for future research.

For cases with available ground truth, traditional statistical metrics like BLEU [39] and ROUGE [40] provide quantitative assessments by comparing generated outputs to reference outputs, focusing on accuracy, structural similarity, and semantic overlap. While these metrics offer precise, objective evaluations, they may not fully capture the complexity and nuances of real-world applications. To address this, integrating these statistical scorers with more advanced methods, such as LLM-as-a-Judge, enables a comprehensive evaluation that combines quantitative precision with deeper qualitative insights, ensuring a robust assessment of model performance.

III. Step Chain of Thought and Task Alignment

To enhance LLMs' semantic understanding and instruction-following, our methodology consists of three stages: A) synthetic data generation, B) model enhancement, and C) model evaluation, as shown in Figure 3. We focus on two tasks:

- **Improving Semantic Understanding in Chatbots**: Using DPO with LoRA and SCoT reasoning to refine semantic interpretation.
- **Aligning User Instructions with Robotics Tasks**: Translating user instructions into task-oriented JSON commands for precise robotic control.

These setups address challenges in interpreting complex instructions. The novelty lies in applying both DPO with LoRA and SCoT to enhance LLM alignment in the research process.

A. Synthetic data generation

To tackle the lack of annotated data, we leverage LLMs to generate high-quality, task-specific synthetic datasets, thereby avoiding the costs and time of manual human annotation and enabling scalable experimentation.

We create two types of datasets: Supervised Fine-Tuning (SFT) datasets and preference datasets.

1) SFT datasets: These provide explicit instructions and corresponding outputs, used to train models with techniques like FFT or LoRA, allowing the model to learn domain-specific knowledge. We create two different SFT datasets:

- Chatbot SCoT Dataset: Contains instructions with corresponding stepwise outputs to enhance chatbots' semantic understanding through SCoT reasoning.
- Robotics JSON Dataset: Contains instructions with JSON outputs, enabling the model to convert natural language instructions into precise commands for robotic control.

Examples of these datasets are depicted in Figure 2, with an additional example in Figure 16 in Appendix.

2) Preference datasets: These datasets are crucial for DPO, as they help align model outputs with human preferences without extensive human feedback. Each dataset contains pairs of outputs for the same instruction: one marked as preferred (chosen (ch.)) and the other as non-preferred (rejected (rej.)),

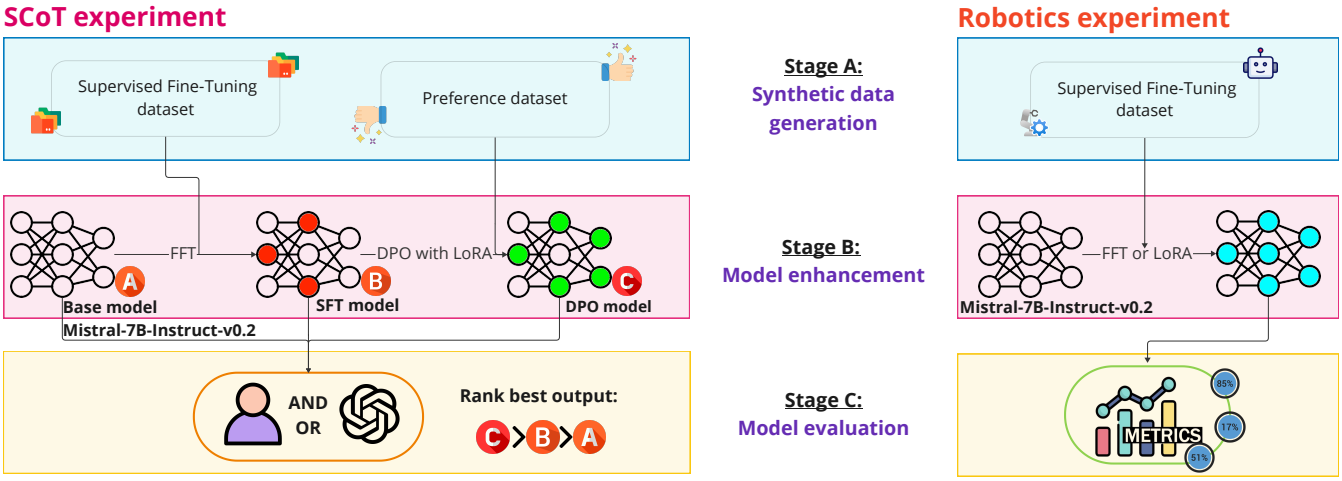


Fig. 3: Simplified overview of the methodology employed in this study, divided into three main stages: Synthetic Data Generation, Model Enhancement, and Model Evaluation.

as illustrated in Figure 4. These datasets train the model to distinguish between higher-quality and lower-quality responses, aligning outputs more closely with human preferences.

To ensure the model prioritizes the best responses, the preference dataset should align more closely with human preferences than the SFT data. To achieve this, we select the ch. responses from models that perform better in this regard than those in the SFT dataset. This approach introduces novel synthetic preference datasets tailored for DPO in the context of SCoT reasoning, as detailed in our contributions.

Preference dataset

Prompt: Below is an original instruction for a task, followed by generated steps that break down the task into smaller, actionable items. Your job is to craft a structured, multi-step prompt for an instruction, ensuring that each step progressively refines the search or selection criteria. Begin with a broad category and incrementally introduce specific filters, retaining ALL previously applied criteria for a focused outcome. Output should present only the steps without the prompt, ensuring each step retains all information from the preceding ones without introducing extraneous details. Number of steps should vary depending on the instruction.

[INST] Convert the instruction prompt to step-wise prompt for: **Design a machine learning algorithm that is mindful of ethical concerns such as transparency. [/INST]**

Chosen (ch.) response	Rejected (rej.) response
1. Identify machine learning algorithms. 2. Narrow down the machine learning algorithms mindful of ethical considerations. 3. Finalize with machine learning algorithms mindful of ethical considerations such as transparency.	1. Identify the type of machine learning algorithm you will design: supervised, unsupervised, or reinforcement learning. 2. Establish a clear objective for your algorithm, ensuring it aligns with ethical considerations. 3. Design the input features for your algorithm, ensuring they are transparent and explainable.

Fig. 4: A sample from the preference dataset used in the SCoT experiments. The ch. response better follows SCoT rules by providing clear, progressive steps.

B. Model enhancement

In the model enhancement stage, Mistral-7B-Instruct-v0.2 [41] is fine-tuned and adapted for specific instruction-following tasks in semantic understanding and robotic instruction translation. We employ several techniques, including SCoT, FFT, LoRA, and DPO with LoRA, each selected to address specific aspects of model alignment and efficiency.

1) SCoT: SCoT is a novel variation of CoT that we introduce to enhance the model's ability to interpret complex instructions. It breaks down complex tasks into smaller, sequential steps, refining the task while remaining consistent with the initial goal. Starting with a broad instruction, each step logically introduces additional details. The final step encapsulates all necessary information without introducing anything new, ensuring a coherent and complete solution. This structured approach improves task precision and helps the model interpret complex instructions more effectively, leading to better alignment with user intent.

2) FFT: FFT involves retraining all weights of the pre-trained LLM for specific downstream tasks. This approach optimizes the model's performance by adjusting its pretrained parameters based on new data. In Equation 1, all model parameters Φ are optimized to maximize the likelihood of generating the correct token sequence y given input tokens x and preceding tokens $y_{<t}$. This equation aims to find Φ that maximizes the log probabilities (logps) of target tokens:

$$\max_{\Phi} \sum_{(x,y)\in\mathcal{Z}} \sum_{t=1}^{|y|} \log\left(P_{\Phi}(y_t|x, y_{<t})\right) \tag{1}$$

While effective, FFT is computationally intensive and may lead to overfitting with limited data.

3) LoRA in PEFT: To address FFT challenges, we use LoRA in PEFT. LoRA introduces trainable low-rank matrices A and B into each Transformer layer, keeping the original

model weights frozen. These matrices provide low-rank updates to the linear layers, as shown in Equation 2, where W_0 is the pretrained weight matrix and ΔW is the low-rank update.

Here, d is the input dimension, k is the output dimension, and $r < \min(d, k)$ is the rank of the low-rank matrices, with $A \in \mathbb{R}^{r \times d}$ and $B \in \mathbb{R}^{k \times r}$. A projects x from d to r, and B then projects it to k. This low-rank update ΔW enhances adaptability while conserving computational resources.

$$h = W_0 x + \Delta W x = W_0 x + BAx \tag{2}$$

4) DPO: DPO utilizes two models, the trained LLM, which undergoes optimization, and a duplicate that is frozen to serve as the reference model during fine-tuning. For each data point, both models generate scores for the ch. response (y_w) and the rej. response (y_l), based on token-level probabilities. The expectation is over the dataset $\mathcal{D}$. The DPO loss function maximizes the likelihood of the ch. response relative to the rej. response by comparing the output probabilities of the trained LLM (π_θ) with those from the frozen LLM (π_{ref}), as represented in Equation 3.

The parameter β scales this process, guiding the updates to the trained LLM to align with human preferences while being anchored by the frozen LLM. The loss function uses a logistic sigmoid function, σ, to convert the difference in logps into an optimizable form.

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right] \tag{3}$$

5) DPO with LoRA: To address the repetitive outputs observed with direct DPO, we integrate LoRA into the DPO framework. Direct DPO can lead to overfitting, causing repetitive token generation due to excessive parameter updates. By incorporating LoRA, only the low-rank matrices are trainable, acting as a regularizer that preserves pre-trained knowledge and promotes better generalization.

We hypothesize that this low-rank constraint smoothens the optimization landscape, reducing the risk of the model becoming trapped in sharp minima associated with repetition. This selective adaptation helps maintain diversity and coherence in the generated text while aligning more effectively with preference data. To the best of our knowledge, integrating DPO with LoRA is not widely discussed in formal literature.

C. Model evaluation

This stage validates that the models perform well on training data and generalize effectively to new, unseen data. We measure training loss across all models. For the SCoT experiment, we also calculate logps and reward margins.

Logps indicate the model's confidence in generating a response. For each response, the total logps is calculated by summing the logps of each token in the sequence, with each token's logps contributing to the total.

As shown in Equation 4, the total logps for a sequence y_i given input x_i is calculated as:

$$\log \pi(y_i \mid x_i) = \sum_{j=1}^{n} \log \pi(y_{ij} \mid x_i) \tag{4}$$

A higher total log probability indicates greater model confidence in generating the sequence. In the context of preference alignment, a higher logps for the preferred response suggests that the model is prioritizing responses that better align with the human preferences dataset.

Reward Margins in DPO represent the difference between the logps assigned to y_w and y_l, scaled by the parameter β. This margin is calculated as:

$$\text{Reward Margin} = \beta \times \left[\Big(\log \pi_\theta(y_w \mid x) - \log \pi_{\text{ref}}(y_w \mid x) \Big) - \Big(\log \pi_\theta(y_l \mid x) - \log \pi_{\text{ref}}(y_l \mid x) \Big) \right] \tag{5}$$

This compares the logps of both responses from the trained LLM (π_θ) and the frozen LLM (π_{ref}). A higher reward margin indicates a greater difference between the ch. and rej. responses from a probabilistic perspective. This means the model assigns significantly higher probability to the preferred response compared to the rejected one, showing a clear distinction in alignment with human preferences.

For the robotics experiment, we use precise evaluation metrics suitable for JSON data: exact match, Levenshtein distance, Jaccard similarity, and F1 score.

Exact match is a binary metric that checks whether the JSON generated by the LLM is identical to the ground truth value, as defined in Equation 6.

$$\text{Exact Match} = \begin{cases} 1 & \text{if generated JSON is identical to ground truth JSON} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Levenshtein distance is the minimum number of single-character edits (insertions, deletions, or substitutions) needed to transform the generated JSON into the ground truth, measuring their difference as defined in Equation 7.

$$\text{lev}_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0 \\ \min \begin{cases} \text{lev}_{a,b}(i-1, j) + 1, \\ \text{lev}_{a,b}(i, j-1) + 1, \\ \text{lev}_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise} \end{cases} \tag{7}$$

Jaccard similarity measures similarity between key-value pairs in the generated JSON and the ground truth, as shown in Equation 8. It calculates the ratio of shared key-value pairs $|A \cap B|$ to the total unique pairs $|A \cup B|$ across both JSONs.

$$\text{Jaccard Similarity} = \frac{|A \cap B|}{|A \cup B|} \tag{8}$$

F1 score combines the precision and recall into one metric:

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

These metrics collectively provide a nuanced assessment of the model's ability to generate accurate and reliable JSON commands in the robotics experiments: Exact Match assesses strict correctness, Levenshtein Distance quantifies the error magnitude, Jaccard Similarity measures semantic overlap, and F1 Score balances precision and recall.

IV. Experimental Setup

The research addresses the primary question: ***"How can Fine-Tuning Techniques be effectively adapted to align LLMs when tasked with generating instructions?"***. To explore this, we conduct two experiments: (1) SCoT Experiments for IKEA's Chatbot and (2) Robotics Experiments for controlling a robotic arm in a grocery-picking scenario. The SCoT experiments use FFT, DPO with LoRA to align LLMs, while the Robotics experiments focus on FFT and LoRA.

A. SCoT Experiments for IKEA's Chatbot

The SCoT experiments aim to improve IKEA's chatbot by structuring instructions into multi-step, task-specific prompts, addressing the main research question and these sub-questions:

- ***"What impact does data quality have on the effectiveness of DPO?"***
  Dataset design significantly influences model alignment and performance. Understanding how dataset characteristics affect outcomes is crucial for more effective training.
- ***"How does the hyperparameter β influence DPO?"***
  The hyperparameter β scales the model's original behavior with human preferences. Its impact on fine-tuning responses is crucial for achieving optimal alignment.

1) Dataset Generation: We create two SCoT datasets.

a) SFT Dataset: Generated through:

- **Instruction Generation**: We generate 6115 instructions using GPT-4-0613 with a temperature of 0.7 to ensure a diverse range of outputs. The temperature is a hyperparameter that controls the randomness of the responses. The prompt, detailed in Figure 7 in Appendix, guides this process by combining direct tasks with narratives across various topics to create clear instructional prompts.
- **SCoT Response Generation**: We use Mistral-8x7B [42] to generate SCoT responses for each instruction at a temperature of 0.1, ensuring deterministic outputs. The prompt outlined in Figure 8 in Appendix, which explains SCoT criteria, includes examples, and ensures adherence to SCoT guidelines.

b) Preference dataset: Contains 2,000 samples covering topics distinct from the SFT dataset, constructed through:

- **Instruction Generation**: We generate instructions using GPT-4-0613, employing the same prompt as for the SFT dataset.
- **SCoT Response Generation**: We generate SCoT responses using models such as GPT-4-0613, LLaMa-3 70B [43], GPT-3.5 Turbo, Mistral 8x22B, or Command R+ for each instruction at a temperature of 0.1. The prompt, adapted from the SCoT generation prompt used for the SFT dataset, as shown in Figure 8 in Appendix, is tailored for each model by adjusting examples and refining criteria to optimize SCoT responses.
- **FFT SCoT Response Generation**: We generate responses using a model fine-tuned with the SFT dataset through FFT, following the fine-tuning prompt shown in Figure 11 in Appendix at a temperature of 0.

We combined the SCoT responses and FFT responses to create **four** distinct datasets, evaluating their effect on DPO. We use a LLM-as-a-Judge to select the best and worst responses based on a predefined set of criteria. The prompt used for this process, detailed in Figure 13 in Appendix, includes an instruction, good and bad examples, and evaluation criteria focused on step continuity, keyword consistency, and appropriate detail levels.

- **Dataset A**:
  - **Ch. Responses**: Generated by GPT-4-0613.
  - **Rej. Responses**: Generated by the FFT model.
  - **Purpose**: To assess whether high-quality responses from GPT-4-0613 lead to superior model alignment compared to lower-quality responses from open-source models. This dataset tests if leveraging GPT-4-0613's quality improves overall performance.
- **Dataset B (Baseline)**, **Dataset C**, and **Dataset D**:
  - **Ch. Responses**: Selected by GPT-4-0613 from open-source models (LLaMa-3 70B, Mistral 8x22B, or Command R+) and GPT-3.5 Turbo.
  - **Rej. Responses**:
    - **Dataset B**: Generated by the FFT model.
    - **Dataset C**: Randomly selected from remaining models (excl. the ch. response) as in distilled direct preference optimization (dDPO) [44].
    - **Dataset D**: Selected by GPT-4-0613 as the worst response from the remaining models.
  - **Purpose**:
    - **Dataset B**: Serves as the baseline to evaluate if penalizing the model's own responses is more effective than using external model outputs. We chose Dataset B as it showed the best performance in preliminary tests.
    - **Dataset C**: Explores the impact of the dDPO method, specifically how randomness in selecting rej. responses affects model performance.
    - **Dataset D**: Tests the impact of explicitly penalizing the worst responses, as identified by the Judge, to refine model alignment. It explores whether targeting the least favorable responses leads to more significant improvements in model performance.

Table VI in Appendix details the distributions of all datasets.

2) Model Enhancement: We fine-tuned the Mistral-7B-Instruct-v0.2 using FFT with the SFT dataset. The fine-tuning prompt, shown in Figure 11 in Appendix, includes a

description of SCoT, with instructions placed between [INST] and [\INST] tokens. These control tokens, which the tokenizer does not encode, serve to mark user message boundaries and prevent prompt injection. This is followed by the corresponding SCoT training sample to prompt the model to "Convert the instruction prompt to a step-wise prompt". After FFT, DPO combined with LoRA further refines the model's alignment with the preference datasets.

a) Base Experiment: We evaluated model enhancement by comparing three versions of the model:

- **Base**: Pre-trained Mistral-7B-Instruct-v0.2, used to evaluate the FFT and DPO impact.
- **SFT**: Mistral-7B-Instruct-v0.2 after FFT.
- **DPO**: SFT model refined with DPO with LoRA.

b) Data Quality Experiment: We examined how different configurations of preference datasets (A, B, C, D)—based on the source and selection of ch. and rej. responses—impact DPO performance.

c) β Hyperparameter Experiment: We test various β values to determine their effects on training loss, reward margins, and response quality, aiming to find the optimal setting for model alignment.

The hyperparameters for both FFT and for the DPO experiments are detailed in Table V in Appendix.

3) Evaluation: We evaluate the models using LLM-as-a-Judge and human evaluation.

LLM-as-a-Judge leverages the prompt in Figure 13 in Appendix to assess responses based on step continuity, keyword consistency, final step completeness, initial step detail, and the appropriate number of steps. As of July 2024, ProLLM [45] ranks GPT-4o (accuracy: 0.82) and GPT-4 Turbo (accuracy: 0.85) as the top LLM Judges, so both are selected.

Human evaluation offers a nuanced assessment. Initially, I perform a blind assessment by shuffling the responses, ensuring unbiased evaluation using the LLM-as-a-Judge prompt. For validation, five additional participants assess five samples from each experiment, following the same guidelines.

To measure the model's ability, we use 200 new test instructions generated with GPT-4-0613, applying the same prompt used for the SFT dataset. These test instructions cover topics not included in the training data.

B. Robotics Experiments for Controlling a Robotic Arm

To tackle evaluation challenges and reduce the need for human labeling, the robotics experiment is designed for the robot to perform actions in the correct sequence, allowing direct evaluation through metrics. This setup demonstrates the practical application of LLMs and provides a robust assessment method.

These experiments explore the fine-tuning methodology in robotics, specifically addressing: ***"How can the proposed fine-tuning methodology be applied in robotics?"***. We use FFT and LoRA, excluding DPO, since the focus is on task alignment and control rather than text generation.

1) Dataset Generation: We use the **"LLM robot" dataset** [21] as ground truth. This dataset is used to train LLMs to generate robotic plans, featuring tasks where a robotic arm manipulates objects based on user instructions. Each entry includes an instruction and a corresponding JSON response detailing the robot's low-level functions. To evaluate the impact of different data sources on performance, we use three distinct datasets:

- **Dataset 1 (Ground Truth Data)**: Consists of 2181 samples from the "LLM Robot" dataset.
- **Dataset 2 (Synthetic Data)**: Includes **10318** synthetic samples generated using GPT-4o through a two-step process. First, for each ground truth training sample, we generate approximately five similar instructions using GPT-4o with temperature settings from 0.1 to 0.7 as shown in Figure 9 in Appendix, to introduce creativity while maintaining alignment with the original data. Second, we generate corresponding JSON outputs using GPT-4o at a temperature of 0, as detailed in Figure 10 in Appendix.
- **Dataset 3 (Ground Truth + Synthetic Data)**: Combines both ground truth and synthetic data for a total of **12499** samples, balancing precision and diversity.

The JSON data from these datasets includes key components to control the robot's actions:

- **Actions Array**: Lists tasks for the robotic arm, each with specific commands and parameters.
- **Command**: Specifies the action (e.g., move, move_to, suction_cup, err_msg).
- **Command**: Includes coordinates (x, y, z) and details like suction cup operation, movement direction, and messages, with non-applicable parameters set to null.

In the original dataset, the environment includes specific elements such as a yellow block and a white block. The synthetic data adds new functionalities to improve the robot's behavior and align with human input:

- **Drop-off Zone**: A designated location for placing items, useful for post-pick actions like shipping.
- **Robustness to Coordinates**: Ensures the robot handles minor variations in coordinates (e.g., +122 vs. 122), reflecting typical human input differences.
- **Error Handling for Invalid Inputs**: Returns an err_msg for invalid objects or unsupported actions (e.g., unfeasible rotations), addressing user errors.

2) Model Enhancement: The Mistral-7B-Instruct-v0.2 model is fine-tuned for robotics tasks using the prompt in Figure 12 in Appendix. The prompt includes a description, an instruction enclosed between [INST] and [\INST] tokens, directing the model to "Convert the instruction prompt to a JSON command sequence", followed by the corresponding JSON output.

We explore two fine-tuning approaches to identify the best alignment with task requirements:

- **FFT**: M1 (Dataset 1), M2 (Dataset 2), M3 (Dataset 3).
- **LoRA**: M1-LoRA (Dataset 1), M2-LoRA (Dataset 2), M3-LoRA (Dataset 3).

Hyperparameters for both FFT and LoRA are detailed in Table V in Appendix.

3) Evaluation: We evaluate the robotics experiments using "ground truth testing data" from the "LLM Robot" dataset, which mirrors the training data with slight variations in phrasing (e.g., "Move arm down 3 times" becomes "Move robotic arm down 3 times"). To further assess adaptability and robustness, we introduce an "exploratory dataset" that includes altered instructions, such as requesting movements beyond the trained limits (e.g., "move up and down 8 times" instead of 6) or introducing untrained actions like "jumping" to test error handling. This exploratory data is distinct from the ground truth and training data, ensuring a thorough evaluation.

The evaluation consists of 215 samples from the "LLM Robot" dataset and 30 exploratory samples introducing new functionalities.

With this setup, we aim to enhance LLM adaptability to real-world applications and show how fine-tuning can improve performance in diverse, task-specific contexts.

V. Experimental Results

This section presents the results of the SCoT experiment, as well as the Robotics experiment.

A. SCoT Experiments for Ikea's Chatbot

We begin by evaluating the base performance through fine-tuning techniques before exploring the effects of dataset quality and the impact of the β parameter on DPO. This evaluation is followed by a validation phase with a larger participant pool to ensure the robustness of our findings.

1) Base Experiment: To evaluate the methodology, we analyze the outputs of the base model, the model after FFT, and the model after DPO with LoRA with dataset B. FFT is applied using the SFT dataset specifically generated for the DPO experiments, and this SFT model serves as the foundation for all subsequent DPO experiments. Responses are generated using a fine-tuning prompt that describes SCoT and includes test instructions, as shown in Figure 11 in Appendix.

DPO is performed with LoRA, as typical learning rates (1e-5 to 5e-5) without LoRA cause word/sentence repetition, an example can be seen in Figure 14 in Appendix. To avoid this, learning rates must be significantly reduced (1e-6 to 1e-7), so DPO is only conducted with LoRA.

We employ an LLM-as-a-Judge for the initial assessment. However, the Judge LLM struggles to find a significant difference between SFT and DPO responses when evaluating based on the SCoT criteria, which include step continuity, keyword and detail consistency, final step completeness, initial step detail, and appropriate number of steps, as described in the prompt shown in Figure 13 in Appendix.

As depicted in Table I, GPT-4 Turbo selects SFT as best 46.50% of the time and DPO 44.50%, while GPT-4o prefers SFT 55.50% of the time compared to 43.50% for DPO. Both Judges rarely choose the base model. Given these close results, human evaluation is conducted, showing a preference for DPO 77.00% of the time versus 23.00% for SFT. The analysis shows that SFT responses already meet many SCoT criteria specified in Figure 13 in Appendix.

Additionally, we find that the Judge LLM is highly sensitive to minor changes in the prompt, which can result in the LLM selecting a different model as the best. This underscores the importance of human evaluation for more accurate assessments, as LLMs may struggle with nuanced evaluations.

DPO aims to align models with human preferences, and the human evaluation results show that the DPO with LoRA model is more effective in meeting this goal.

Evaluation Metric	Base	SFT	DPO
LLM-as-a-Judge (GPT-4o)			
Chosen as best (%)	1.00	**55.50**	43.50
LLM-as-a-Judge (GPT-4-Turbo)			
Chosen as best (%)	9.00	**46.50**	44.50
Human Evaluation			
Chosen as best (%)	0.00	23.00	**77.00**

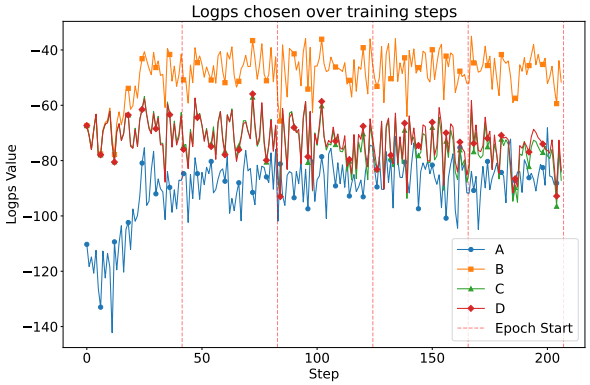
TABLE I: Evaluation results for LLM-as-a-Judge and human evaluations by a single participant for 200 test data samples generated at temperature 0.

2) Dataset Quality Experiment: To evaluate the effect of dataset quality on DPO, we analyze reward margins, logps, and human evaluation results across four datasets (A, B, C, and D). We exclude the LLM-as-a-Judge since it could not effectively distinguish between the SFT and DPO models. The DPO models are more similar to each other than to the SFT model, differing only by preference data and not by hyperparameters, as depicted in Table V in Appendix, making LLM-as-a-Judge less useful for this comparison.

Our human evaluation results, summarized in Table II, show that Dataset B is chosen as the best 70% of the time, reflecting its superior alignment. In comparison, Dataset A is chosen 15% of the time, Dataset C 10%, and Dataset D 5%.

The analysis of logps for ch. responses, illustrated in Figure 5a, shows that Dataset B has the highest logps, indicating better model confidence and alignment. Although Dataset A also shows an increase, it remains lower than Dataset B. This may be due to Dataset B consisting of 76.15% open-source models (Mistral-8x22B, LLaMa-3 70B, Command R+), as detailed in Table VI in Appendix. These models likely share pre-training data with the Mistral-8x7B used for the SFT model. In contrast, Dataset A contains data from GPT-4-0613, a closed-source model likely trained on different data, leading to lower alignment with the SFT model. This emphasizes the importance of choosing datasets that are well-aligned with the training data to achieve optimal performance.

In contrast, Datasets A and B show higher logps for rej. responses, Figure 5b, suggesting these rej. responses are more aligned with the frozen LLM's output. In the context of DPO, this higher alignment indicates that the model's confidence in these responses is closer to that of the frozen LLM. Conversely, the lower logps for rejected responses in Datasets



(a) Logps of the ch. responses during training.

(b) Logps of the rej. responses during training.

Fig. 5: Logps for ch. and rej. responses across different datasets during the data quality experiment.

C and D imply a divergence from the frozen LLM's behavior, suggesting these responses would not have been generated by the frozen LLM.

Using the SFT model's response as the rej. option in these datasets is crucial for achieving higher logps and better performance. The consistently higher logps in Dataset B, for both ch. and rej. responses, drive its superior performance, as confirmed by human evaluations.

Although Dataset A demonstrates the highest reward margins throughout the training steps, shown in Figure 17 in Appendix, this does not translate into the best performance, as evidenced by the human evaluations. These findings suggest that logps are a more critical factor than overall reward margins in determining the model's performance.

	A	B	C	D
Human Evaluation				
Chosen as best (%)	15.00	**70.0**	10.00	5.00

TABLE II: Human evaluations for 20 test data samples generated at temperature 0 for dataset quality experiment.

3) β experiment: This experiment aims to investigate the impact of different β values on the DPO process and identify the optimal β for aligning model outputs with human preferences in SCoT using Dataset B.

As shown in Figure 19 in Appendix, higher β values (e.g., $\beta = 1$) result in lower loss, higher reward margins, and a smaller decrease in the logps of rej. responses, whereas lower β values (e.g., β = 0.2) show the opposite trend.

For each test instruction in the human evaluation, we select the best model and the two worst models. As summarized in Table III, the model with $\beta = 0.2$ is chosen as the best 45% of the time and the worst only 2.5% of the time. The model with $\beta = 0.5$ is chosen as the best 30% of the time but is also selected as the worst 17.5% of the time, indicating higher performance variability. Models with higher β values ($\beta = 0.7$ and $\beta = 1$) are more frequently chosen as the worst, with $\beta = 1$ being the worst 42.5% of the time and never chosen as the best.

Qualitative analysis shows that there is often similarity between two or three responses of the worst-performing models, leading to the selection of the two worst models rather than just one. Additionally, very low β values (0.01, 0.1) produce repetitive phrases or sentence structures, as seen in Figure 14 in Appendix. The rewards for the ch. responses, as illustrated in Figure 18 in Appendix, become negative through the training. We hypothesize that these negative rewards for ch. responses lead to repetition.

To contextualize our findings within existing research, it is notable that a β value of 0.1 is commonly used [19, 43], with studies exploring a range between 0.01 and 0.5 [46]. As shown in Figure 15 in Appendix, the $\beta = 0.2$ model eliminates repetitions, while the $\beta = 0.1$ model exhibits them. Our results, which involve DPO with LoRA, align with these findings, showing that lower β values, particularly around 0.2 and 0.3, effectively minimize poor responses while maintaining alignment with human preferences. This suggests that while $\beta = 0.1$ is a widely accepted and effective choice in traditional DPO setups, exploring higher values like 0.2 or 0.3 within the DPO with LoRA framework may provide a more stable balance between response quality and model reliability. Thus, our study offers nuanced insights into optimizing β for fine-tuning models using DPO with LoRA.

Our results are based on specific hyperparameters, where only the beta value is varied while other factors are constant. Therefore, our findings should be interpreted with this constraint in mind, as we did not explore a broad sweep of hyperparameters in combination with β. This suggests that while our insights are valuable for understanding the role of β in DPO, further research is needed to explore how β interacts with a wider range of hyperparameters.

4) Human validation: To validate the initial experiments, we conduct additional evaluations with five individuals. In the base experiment, 64% of the DPO samples are rated as the best, aligning with the previous result of 77%, as can be seen in Table VII in Appendix. In the dataset quality experiment,

Metric	$\beta = 0.2$	$\beta = 0.3$	$\beta = 0.5$	$\beta = 0.7$	$\beta = 1$
Human Evaluation					
Chosen as best (%)	**45.00**	20.00	30.00	5.00	0.00
Chosen as worst (%)	2.50	2.50	17.50	35.00	**42.50**

TABLE III: Human evaluations for models with different β values on 20 test data samples generated at temperature 0. For each test instruction, the best model and the two worst models are selected.

68% of samples favored Dataset B, consistent with the initial result of 70%, as shown in Table VIII in Appendix.

In the β experimentation, the evaluations reveal more variation in preferences. While $\beta = 1$ is still not chosen as the best, some participants preferred the model with $\beta = 0.7$, as shown in Table IX in Appendix, indicating that personal preference plays a significant role in model evaluation. This highlights the importance of considering the user-specific preferences when fine-tuning models.

However, with only five participants in the validation phase and six in total including the initial experiment, the statistical significance of these results is limited. Further validation with a larger and more diverse participant pool is necessary to ensure the reliability of these findings, especially in the β experimentation, where variability in preferences is observed.

B. Robotics Experiments for Controlling a Robotic Arm

We evaluate three models —M1, M2, and M3—trained using either FFT or LoRA, on both ground truth testing data and exploratory data introducing new functionalities. As detailed in Table IV, M1, trained exclusively on ground truth data using FFT, achieves the highest scores on ground truth testing data with exact match (0.99), Jaccard similarity (0.99), and F1 score (0.99), indicating strong accuracy and alignment with expected outputs. M3, trained on a combination of ground truth and synthetic data, also performs well, exhibiting the lowest Levenshtein distance (0.15), suggesting minimal edits are needed to match the ground truth.

In contrast, M2, trained solely on synthetic data, significantly underperforms compared to M1 and M3 on ground truth testing data, which, despite being generated with detailed prompts using GPT-4o, as seen in Figure 10 in Appendix, only achieved 98.1% accuracy on the training data. The small inaccuracies introduced impacted the model's overall performance. While augmenting ground truth data with synthetic data (as in M3) improved performance to levels comparable with M1, relying solely on synthetic data (as in M2) is not recommended due to inherent inaccuracies.

When tested on exploratory data with new functionalities, models trained with LoRA—specifically M2-LoRA and M3-LoRA—exhibited zero-shot capabilities and superior performance, particularly in adapting to novel instructions. As shown in Table IV, M2-LoRA achieved the lowest Levenshtein distance (24.79) and the highest Jaccard similarity (0.74) and F1 score (0.72). This improvement is likely due to LoRA's regularization effect, which helps maintain the base model's performance across a broader range of tasks by "learning less and forgetting less" [33]. Notably, M2-LoRA outperformed its fully fine-tuned counterpart (M2) across all metrics, benefiting from LoRA's enhanced generalization to novel instructions.

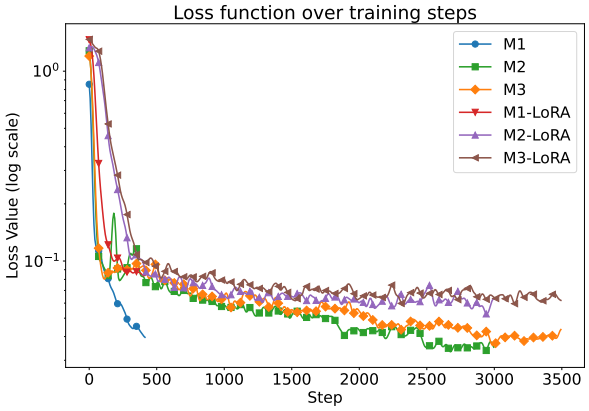


Fig. 6: The loss function over training steps for the robotics experiment, Gaussian filter applied to improve the readability of the plot. Note that training steps vary due to the differing amounts of data per dataset.

This generalization capability is further supported by the training loss convergence patterns observed in Figure 6. Specifically, M1-LoRA, M2-LoRA, and M3-LoRA exhibit higher loss values than M1, M2, and M3, respectively. This higher loss convergence may be attributed to LoRA's regularization properties, which prevent overfitting to the training data and enable broader applicability.

Qualitative analysis reveals distinct error patterns between models trained with FFT and LoRA. FFT-trained models like M1 occasionally produced errors in JSON structure and parameter alignment, particularly with complex commands. M2, trained solely on synthetic data, frequently introduced incorrect parameters and faulty JSON structures, likely due to overfitting to less accurate synthetic examples. M3, combining ground truth and synthetic data, showed improved handling of trained commands, but still exhibited minor inaccuracies, such as rounding errors in numerical values.

LoRA-trained models, while generally more adaptable, displayed unique errors, such as generating untrained commands like "say" or "shutoff." These hallucinations suggest that LoRA's regularization can sometimes lead to overgeneralization. Notably, M3-LoRA occasionally repeated commands, contributing to higher Levenshtein distances. Despite these issues, LoRA-enhanced models demonstrate better generaliza-

Metric	M1	M2	M3	M1-LoRA	M2-LoRA	M3-LoRA
On Ground Truth Testing Data						
Exact Match	**0.99**	0.80	0.96	0.98	0.86	0.91
Levenshtein Distance	0.16	5.87	**0.15**	4.45	1.42	1.55
Jaccard Similarity	**0.99**	0.90	0.97	**0.99**	0.90	0.95
F1 Score	**0.99**	0.90	0.98	**0.99**	0.92	0.96
On Exploratory Data with New Functionalities						
Exact Match	0.07	0.37	0.53	0.23	**0.67**	**0.67**
Levenshtein Distance	141.71	82.24	51.30	66.86	**24.79**	66.17
Jaccard Similarity	0.18	0.47	0.60	0.28	**0.74**	0.71
F1 Score	0.17	0.47	0.59	0.28	**0.72**	0.70

TABLE IV: Metrics evaluated at temperature 0 for various model setups on 215 ground truth testing samples and 30 exploratory testing samples.

tion to novel tasks, but they also propagate common error patterns due to their tendency to generate untrained or unexpected commands across different datasets.

Overall, our results suggest that while FFT achieves optimal performance on familiar tasks, LoRA significantly enhances the model's adaptability to new and varied scenarios. A key finding is the importance of data diversification: combining ground truth and synthetic data, particularly when paired with LoRA, provides a balanced training approach that ensures both accuracy and flexibility. This diversified dataset allows the model to effectively process both known and novel tasks, making it more robust and versatile in handling a wide range of robotic functions.

VI. Conclusion

In this study, we explore methods to align LLMs by applying DPO, LoRA, and FFT for SCoT and robotic task execution. Our approach consists of three stages: synthetic data generation, model enhancement, and model evaluation. The goal is to adapt LLMs to respond more effectively to complex, real-world tasks.

A key component is the introduction of SCoT, which improves the processing of user instructions by breaking tasks into structured steps. This approach enables LLMs, like the IKEA chatbot, to enhance semantic understanding and produce more accurate, contextually relevant responses.

Through DPO experiments, we demonstrate that combining DPO with LoRA leads to greater alignment with human preferences than FFT alone. The DPO models consistently outperformed base and SFT models in human evaluations, indicating stronger alignment. We also find that DPO model performance heavily depends on preference dataset quality, with well-aligned datasets yielding better results. Lower β values during DPO training help maintain alignment, but require careful tuning to avoid repetition in outputs.

In the robotics experiment, we evaluate models trained on ground truth data, synthetic data, and a combination of both, highlighting the importance of data diversification. LoRA-trained models, especially those utilizing both data types, demonstrated superior zero-shot capabilities and adaptability to unseen tasks compared to FFT-trained models. LoRA's regularization effect helps mitigate overfitting, making it more effective in generalizing to new challenges despite higher loss values during training. While FFT excels on tasks similar to the training data, LoRA consistently outperformed it in diverse, real-world applications.

Overall, integrating LoRA into model training significantly enhances the model's ability to generalize to new and varied instructions, which is critical for practical applications like robotic arm control. For the IKEA chatbot, the use of SCoT and DPO with LoRA resulted in responses that align more closely with human preferences, aiming to improve user experience. In robotics, LoRA-enhanced models contribute to more flexible robotic systems capable of adapting to dynamic environments, reducing the need for retraining when introducing new tasks and increasing operational efficiency.

VII. Future work

This study suggests several future research directions to enhance LLMs' task alignment. One promising area is refining the SCoT process, particularly for instruction-based LLMs like the IKEA chatbot. Investigating whether incorporating SCoT improves task execution and user interaction would be a valuable next step.

Additionally, addressing repetition in DPO is crucial. This study shows that DPO combined with LoRA is more effective, but the underlying reasons remain unclear. Future research could explore techniques like masking special formatting tokens in the DPO loss calculation, similar to the approach in Llama 3 [43].

Incorporating RLHF as a benchmark could provide deeper insights into the strengths and weaknesses of DPO and other alignment strategies.

Improving the LLM-as-a-Judge approach is another important research avenue. Fine-tuning models specifically as evaluators, as suggested by [38], could enhance their ability to assess complex outputs like SCoT. Additionally, the Panel of LLM Evaluators (PoLL) [47], where multiple smaller models collectively score an answer through a voting function, could offer a more reliable evaluation framework.

Future research could explore the effects of varying ranks in LoRA. Additionally, newer LoRA derivatives like DoRA

(Weight-Decomposed Low-Rank Adaptation) [48] could be investigated for their potential benefits.

Finally, testing with real human instructions, rather than synthetic data, could provide valuable insights into the model's ability to generalize to authentic user inputs. This approach would also help assess the model's practical applicability in real-world robotic tasks.

Acknowledgment

I would like to thank my supervisors and the AI Lab team for their support and guidance throughout this research. A special thanks to the fellow students at RoboHouse for their insightful discussions and the enjoyable time working together. I am also grateful to the five human labelers who helped evaluate my SCoT models and to my friends that reviewed this paper.

References

[1] W. Yu *et al.*, *Language to Rewards for Robotic Skill Synthesis*, arXiv:2306.08647 [cs], Jun. 2023. [Online]. Available: http://arxiv.org/abs/2306.08647 (visited on 03/14/2024).

[2] D. Han, T. McInroe, A. Jelley, S. V. Albrecht, P. Bell, and A. Storkey, *Llm-personalize: Aligning llm planners with human preferences via reinforced self-training for housekeeping robots*, 2024. arXiv: 2404.14285 [cs.RO].

[3] D. Driess *et al.*, *PaLM-E: An Embodied Multimodal Language Model*, arXiv:2303.03378 [cs], Mar. 2023. [Online]. Available: http://arxiv.org/abs/2303.03378 (visited on 03/14/2024).

[4] D. Shah, B. Osiński, S. Levine, *et al.*, *Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action*, PMLR, 2023.

[5] C. Wang *et al.*, "Lami: Large language models for multi-modal human-robot interaction," in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–10.

[6] M. Cheung, "A reality check of the benefits of llm in business," *arXiv preprint arXiv:2406.10249*, 2024.

[7] S. G. Ayyamperumal and L. Ge, "Current state of llm risks and ai guardrails," *arXiv preprint arXiv:2406.12934*, 2024.

[8] W. Liu *et al.*, "Aligning large language models with human preferences through representation engineering," *arXiv preprint arXiv:2312.15997*, 2023.

[9] Y. Wolf, N. Wies, O. Avnery, Y. Levine, and A. Shashua, "Fundamental limitations of alignment in large language models," *arXiv preprint arXiv:2304.11082*, 2023.

[10] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in neural information processing systems*, vol. 30, 2017.

[11] N. Stiennon *et al.*, *Learning to summarize from human feedback*, arXiv:2009.01325 [cs], Feb. 2022. [Online]. Available: http://arxiv.org/abs/2009.01325 (visited on 03/11/2024).

[12] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[13] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[14] M. Spahn *et al.*, "Demonstrating adaptive mobile manipulation in retail environments," in *Robotics: Science and Systems (R:SS)*, 2024.

[15] J. Wei *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[16] B. Chen, Z. Zhang, N. Langrené, and S. Zhu, *Unleashing the potential of prompt engineering in Large Language Models: A comprehensive review*, arXiv:2310.14735 [cs], Oct. 2023. [Online]. Available: http://arxiv.org/abs/2310.14735 (visited on 03/27/2024).

[17] OpenAI, *Openai o1 system card*, 2024. [Online]. Available: https://cdn.openai.com/o1-system-card.pdf.

[18] E. J. Hu *et al.*, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9.

[19] T. Widatalla, R. Rafailov, and B. Hie, "Aligning protein generative models with experimental fitness via direct preference optimization," *bioRxiv*, 2024. DOI: 10.1101/2024.05.20.595026. [Online]. Available: https://www.biorxiv.org/content/early/2024/05/21/2024.05.20.595026.

[20] L. Zheng *et al.*, "Judging llm-as-a-judge with mt-bench and chatbot arena," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[21] Aryaduta, *Llm_robot*, 2024. [Online]. Available: https://huggingface.co/datasets/Aryaduta/llm_robot.

[22] F. Zeng, W. Gan, Y. Wang, N. Liu, and P. S. Yu, *Large Language Models for Robotics: A Survey*, arXiv:2311.07226 [cs], Nov. 2023. [Online]. Available: http://arxiv.org/abs/2311.07226 (visited on 03/25/2024).

[23] C. Zhang, J. Chen, J. Li, Y. Peng, and Z. Mao, "Large language models for human-robot interaction: A review," *Biomimetic Intelligence and Robotics*, p. 100 131, 2023.

[24] S. Tellex, N. Gopalan, H. Kress-Gazit, and C. Matuszek, "Robots that use language," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 25–55, 2020.

[25] b. ichter brian *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *Proceedings*

of The 6th Conference on Robot Learning, K. Liu, D. Kulic, and J. Ichnowski, Eds., ser. Proceedings of Machine Learning Research, vol. 205, PMLR, 14–18 Dec 2023, pp. 287–318. [Online]. Available: https://proceedings.mlr.press/v205/ichter23a.html.

[26] A. Q. Jiang *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[27] Y. Mu *et al.*, "Embodiedgpt: Vision-language pre-training via embodied chain of thought," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[28] X. L. Li and P. Liang, *Prefix-Tuning: Optimizing Continuous Prompts for Generation*, arXiv:2101.00190 [cs], Jan. 2021. [Online]. Available: http://arxiv.org/abs/2101.00190 (visited on 02/22/2024).

[29] J. Xiang *et al.*, "Language models meet world models: Embodied experiences enhance language models," *Advances in neural information processing systems*, vol. 36, 2024.

[30] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[31] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*, vol. 24, Elsevier, 1989, pp. 109–165.

[32] X. Meng *et al.*, *Periodiclora: Breaking the low-rank bottleneck in lora optimization*, 2024. arXiv: 2402.16141 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2402.16141.

[33] D. Biderman *et al.*, "Lora learns less and forgets less," *arXiv preprint arXiv:2405.09673*, 2024.

[34] J. Wu *et al.*, "β-dpo: Direct preference optimization with dynamic β," *arXiv preprint arXiv:2407.08639*, 2024.

[35] B. Qi, P. Li, F. Li, J. Gao, K. Zhang, and B. Zhou, "Online dpo: Online direct preference optimization with fast-slow chasing," *arXiv preprint arXiv:2406.05534*, 2024.

[36] OpenAI *et al.*, *GPT-4 Technical Report*, arXiv:2303.08774 [cs], Mar. 2024. [Online]. Available: http://arxiv.org/abs/2303.08774 (visited on 03/27/2024).

[37] X. Li *et al.*, *Alpacaeval: An automatic evaluator of instruction-following models*, https://github.com/tatsu-lab/alpaca_eval, May 2023.

[38] H. Huang, Y. Qu, J. Liu, M. Yang, and T. Zhao, "An empirical study of llm-as-a-judge for llm evaluation: Fine-tuned judge models are task-specific classifiers," *arXiv preprint arXiv:2403.02839*, 2024.

[39] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[40] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[41] A. Q. Jiang *et al.*, *Mistral 7B*, arXiv:2310.06825 [cs], Oct. 2023. [Online]. Available: http://arxiv.org/abs/2310.06825 (visited on 02/08/2024).

[42] A. Q. Jiang *et al.*, *Mixtral of Experts*, arXiv:2401.04088 [cs], Jan. 2024. [Online]. Available: http://arxiv.org/abs/2401.04088 (visited on 02/08/2024).

[43] A. Dubey *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[44] L. Tunstall *et al.*, "Zephyr: Direct distillation of LM alignment," in *First Conference on Language Modeling*, 2024. [Online]. Available: https://openreview.net/forum?id=aKkAwZB6JV.

[45] ProLLM, 2024. [Online]. Available: https://prollm.toqan.ai/leaderboard/llm-as-judge.

[46] R. Rafailov *et al.*, "Scaling laws for reward model overoptimization in direct alignment algorithms," *arXiv preprint arXiv:2406.02900*, 2024.

[47] P. Verga *et al.*, "Replacing judges with juries: Evaluating llm generations with a panel of diverse models," *arXiv preprint arXiv:2404.18796*, 2024.

[48] S.-Y. Liu *et al.*, *DoRA: Weight-Decomposed Low-Rank Adaptation*, arXiv:2402.09353 [cs], Mar. 2024. [Online]. Available: http://arxiv.org/abs/2402.09353 (visited on 03/08/2024).

Instruction Generation Prompt for DPO experiment

Crafting Mixed Instructional Prompts
Objective: Create a series of instructional prompts that blend straightforward tasks with narrative-driven scenarios, suitable for breaking down into 2 to 4 actionable steps. Aim for clarity and precision, avoiding any repetition of words and steering clear of questions to ensure straightforward directives.
Key Principles:

- **Direct Instructions:** All prompts should clearly state the task at hand, avoiding interrogative formats to ensure directness and actionability.
- **Varied Contexts:** Craft prompts that stand alone or include a short narrative, ensuring any context provided is supplementary and not repeated in the step-wise breakdown.
- **Clarity and Specificity:** Embed precise criteria within each prompt, facilitating a targeted, step-wise exploration without compromising on clarity.
- **Topic Selection:** <topic>.

Instructional Prompt Examples:

1) Outline the process for conducting a literature review in Quantum Physics focusing on foundational theories.
2) I am going to have a baby, find a dark wooden bed.
3) Set up a small, efficient workspace in a shared apartment, considering ergonomics and productivity.
4) With a family vacation to Europe coming up, plan an itinerary covering historical landmarks and local cuisines.

Request: Please provide 10 additional instructional prompts on the specified topic mentioned above. The prompts should be able to be broken into a maximum of 4 steps, adhering to the outlined principles and ensuring a mix of direct and narrative-driven formats. Exclude step-wise breakdowns and focus solely on the instructions (1 to 10).

Fig. 7: Prompt used for instruction generation for the DPO experiment.

SCoT generation for DPO experiment

Instruction:
Creating Step-Wise Prompts for <instruction>

Objective:
Craft a structured, multi-step prompt for <instruction>, ensuring that each step progressively refines the search or selection criteria. Begin with a broad category and incrementally introduce specific filters, retaining **all** previously applied criteria for a focused outcome.

Instruction Prompt Creation Process:

1) **Start Broad:** Begin with the most general category relevant to your prompt. For example, if your prompt is "Find a yellow wooden bed for two people below 200 euros," your first step is to "Identify the beds."
2) **Add Specifics:** Introduce a new, specific filter in each subsequent step. In the given example, the second step would be to "Narrow down to beds made of wood that are yellow."
3) **Continue Refinement:** Keep adding layers of specificity. The third step for the bed prompt would be to "Select only the yellow wooden beds designed for two people."
4) **Finalize Criteria:** Apply the last filter to achieve the desired specificity, such as "Choose the two-person wooden beds that are yellow and below 200 euros."

Repeat this process for any topic by adapting the initial broad category and subsequent filters to fit the new context.
Additional Example:

- Plan and execute a professional lighting scheme for a dramatic theater scene, taking into consideration mood, visibility, and safety.
    1) Identify a professional lighting scheme.
    2) Select a professional lighting scheme for a dramatic theater scene.
    3) Choose a professional lighting scheme for a dramatic theater scene that takes into consideration the mood.
    4) Narrow to a professional lighting scheme for a dramatic theater scene that takes into consideration the mood and visibility.
    5) Finalize to a professional lighting scheme for a dramatic theater scene that takes into consideration the mood, visibility, and safety.

Output should present only the steps without the prompt, ensuring each step retains all information from the preceding ones without introducing extraneous details. The number of steps should vary depending on the instruction.

Your task is to meticulously follow these steps, ensuring each response is comprehensive, brief, coherent, and builds directly on the information provided in the previous step without introducing information not in the instruction. Think step by step, employing affirmative language to maintain clarity and directness.

Fig. 8: Prompt used for SCoT generation.

Instruction generation prompt for Robotics

Task:
Generate a new instruction based on the following example: <instruction>
Ensure the new instruction is unique but maintains the same style and context related to the robotic arm and object manipulation. The robotic arm is equipped with a suction cup, and the arm can move up, down, left, right, forward, and backward. An action can be repeated more than 5 times up to 15 times.
Blocks' origin coordinates are known to the robot, so the only possibility is to move the block to a specific place—pick-up location is known. To move a block, the suction must be activated first. You can also move a block behind another block, following this logic: "Go to box A — activate suction — go to box B — place box A behind B." Alternatively, a block can be placed on top of another. Blocks can also be placed at a drop-off zone.
The blocks are white or yellow, and coordinates can be given in the following formats:

- (215.45, 150.32, -30.50)
- (+215.45, 150.32, -30.50)
- (215.45, +150.32, -30.50)
- (+215.45, +150.32, -30.50)

Output: The output should be only the new instruction.

Fig. 9: Prompt used for instruction generation.

JSON generation prompt

Given a textual command directed at a robotic arm, you must generate a structured JSON command sequence that the robot arm should follow to accomplish the task described.

1) **Actions Array**: This is where you list all the tasks (or actions) the robotic arm needs to perform. Each task is represented as an object with specific commands and parameters.
2) **Command**: This part of each action specifies what the robot should do. Common commands are `move`, `move_to`, `suction_cup`, and `err_msg`.
3) **Parameters**: These are the details that explain how to execute the command:
   - **x, y, z**: These are coordinates that tell the robot where to go. If the task doesn't need a specific location, you set these to `null`.
   - **action**: This specifies operations like turning the suction (vacuum) cup `on` or `off`. If it's not needed for the command, it's set to `null`.
   - **direction**: This tells the robot which way to move, like `up`, `down`, `right`, `left`, `forward`, `backward`. If no specific direction is needed, this is set to `null`.
   - **msg**: Used for sending messages, especially if there's an error or a special note about the task. If there's nothing to say, this is set to `null`.

Each action should include all these parameters, even if you set them to `null`, to ensure the robot understands exactly what to do.
Drop-off zone is at (100, 50, 25) aka (`"x": 100, "y": 50, "z": 25`). Yellow block is at (249.62, 137.63, -55) aka (`"x": 249.62, "y": 137.63, "z": -55`), white block is at (266.05, 8.32, -53.46) (`"x": 266.05, "y": 8.32, "z": -53.46`). There are no other color blocks or objects, if another object is in the instruction, the action command must be an `err_msg`. If no color is given for the block, the action command must be an `err_msg`.
The robotic arm does not rotate, if the instruction is rotate, the action command must be an `err_msg`.
The coordinates (215.45, 150.32, -30.50), (+215.45, 150.32, -30.50), (215.45, +150.32, -30.50) and (+215.45, +150.32, -30.50) should be converted to (`"x": 215.45, "y": 150.32, "z": -30.50`) in the JSON.
The instruction could also give wrong coordinates for the drop-off zone, yellow block, or white block; in this case, the action command must be an `err_msg`. For unknown commands, the action command must be an `err_msg`.

A. Examples for Reference

- **Example 1**:
  - **Instruction**: "Activate suction, proceed to location (192.72, +229.45, -61.07), go upwards, then deactivate suction."
  - **Actions**:

    { "actions": [{ "command": "suction_cup", "parameters": { "x": null, "y": null, "z": null, "action": "on", "direction": null, "msg": null } }, { "command": "move_to", "parameters": { "x": 192.72, "y": 229.45, "z": -61.07, "action": null, "direction": null, "msg": null } }, { "command": "move", "parameters": { "x": null, "y": null, "z": null, "action": null, "direction": "up", "msg": null } }, { "command": "suction_cup", "parameters": { "x": null, "y": null, "z": null, "action": "off", "direction": null, "msg": null } }] }

  .....(Examples 2 to 25 not shown due to lack of space).....

- **Example 26**:
  - **Instruction**: "Shift towards the left and then rise."
  - **Actions**:

    { "actions": [{ "command": "move", "parameters": { "x": null, "y": null, "z": null, "action": null, "direction": "left", "msg": null } }, { "command": "move", "parameters": { "x": null, "y": null, "z": null, "action": null, "direction": "up", "msg": null } }] }

Generate a JSON based on the following example: <instruction>

Output should be only a JSON, nothing else (just dict).

Fig. 10: Prompt used for the generation of JSON data based on an instruction.

Fine-Tuning Prompt for SCoT experiment

Below is an original instruction for a task, followed by generated steps that break down the task into smaller, actionable items. Your job is to craft a structured, multi-step prompt for an instruction, ensuring that each step progressively refines the search or selection criteria. Begin with a broad category and incrementally introduce specific filters, retaining ALL previously applied criteria for a focused outcome. Output should present only the steps without the prompt, ensuring each step retains all information from the preceding ones without introducing extraneous details. Number of steps should vary depending on the instruction.

[INST] Convert the instruction prompt to step-wise prompt for: **instruction** **[\INST]**

Fig. 11: Fine-Tuning Prompt for SCoT experiment.

Fine-Tuning Prompt for Robotics experiment

[INST] Given a textual command directed at a robotic arm, you must generate a structured JSON command sequence that the robot arm should follow to accomplish the task described. Convert to the corresponding JSON format the instruction: **instruction** **[\INST]**

Fig. 12: Fine-Tuning Prompt for Robotics experiment.

Parameter	Base FFT	Base DPO	Quality Exp.	Beta Exp.	Robotics Exp. (LoRA)	Robotics Exp. (No LoRA)
model_id	Mistral-7B-Instruct-v0.2	Base FFT	Base FFT	Base FFT	Mistral-7B-Instruct-v0.2	Mistral-7B-Instruct-v0.2
accelerator_config	dispatch_batches					
even_batches	true					
optim	adamw_torch					
adam_beta1	0.9					
adam_beta2	0.999					
adam_epsilon	1e-8					
bf16	true					
dataloader_pin_memory	true					
decoder_start_token_id	null					
eos_token_id	2					
eval_do_concat_batches	true					
fp16	false					
hidden_act	silu					
hidden_size	4,096					
length_column_name	length					
log_level	passive					
lr_scheduler_type	cosine					
max_grad_norm	1					
max_position_embeddings	32,768					
num_attention_heads	32					
num_hidden_layers	32					
repetition_penalty	1					
seed	42					
train	true					
vocab_size	32,000					
weight_decay	0					
learning_rate	0.00002	0.00001	0.00002	0.00001	0.00002	0.00002
epoch	3	5	5	5	5	5
batch_size	4	4	6	4	2	2
dataset	**SFT dataset**	**B**	**A, B, C, or D**	**B**	**ground truth, synthetic or combined**	
gradient_accumulation_steps	8	4	4	4	6	6
logging_steps	1	2	2	2	1	1
max_length	20	1,024	1,024	1,024	20	20
max_seq_length	520	1,024	1,024	1,024	520	520
num_train_epochs	3	5	5	5	5	5
per_device_train_batch_size	4	4	6	4	2	2
ref_model_mixup_alpha	-	0.9	0.9	0.9	-	-
ref_model_sync_steps	-	64	64	64	-	-
beta	-	0.1	**0.2**	**0.01, 0.1, 0.2, 0.3, 0.5, 0.7, 1**	-	-
lora_alpha	-	16	16	16	-	16
lora_dropout	-	0.1	0.1	0.1	-	0.1
target_modules	-	”gate_proj”, ”down_proj”, ”k_proj”, ”q_proj”, ”v_proj”, ”up_proj”, ”o_proj”				-
peft_type	-	LORA				-
r	-	128				-

TABLE V: Hyperparameters of this research experiments.

LLM-as-a-Judge Evaluation Prompt. Used for Evaluation of the Best or the Worse Model Response

Review responses from different AI models to a specific instruction and determine the [best]/[worst] responses based on their adherence to ideal example responses and specific criteria. [The best response should closely align with the examples in structure, content, and adherence to these criteria.] / [The worst response should deviate significantly from these standards, lacking either in clarity, accuracy, or relevance.]
Criteria for Evaluation:

- **Step Continuity:** Each step should logically follow from the previous one without introducing new keywords or details that are then abandoned.
- **Keyword and Detail Consistency:** No new keywords or unnecessary details should be introduced beyond what is specified in the initial instruction.
- **Final Step Completeness:** The final step should encapsulate all necessary details without introducing new information.
- **Initial Step Detail:** The first step should contain at least two key details, but not more than half of all necessary details.
- **Appropriate Number of Steps:** The number of steps should be sufficient to cover the instruction but not excessive; ideally, the total number of steps should reflect a balance between comprehensiveness and efficiency.

Example Instruction: 'Plan and execute a professional lighting scheme for a dramatic theater scene, taking into consideration mood, visibility, and safety.':

1) Identify a professional lighting scheme.
2) Select a professional lighting scheme for a dramatic theater scene.
3) Choose a professional lighting scheme for a dramatic theater scene which takes into consideration the mood.
4) Narrow to a professional lighting scheme for a dramatic theater scene which takes into consideration the mood and visibility.
5) Finalize to professional lighting scheme for a dramatic theater scene which takes into consideration the mood, visibility, and safety.

Bad Response Example: 'Given the rise of AI in healthcare, devise a plan to ensure the ethical use of AI in diagnosing and treating patients.'

1) Research the current state of AI in healthcare, including its applications in diagnosing and treating patients.
2) Identify ethical concerns related to the use of AI in healthcare, such as privacy, bias, and informed consent.
3) Review relevant ethical frameworks and guidelines for the use of AI in healthcare.
4) Evaluate existing AI systems in healthcare against these ethical frameworks and guidelines.
5) Develop a checklist or rubric for assessing the ethical implications of new AI systems in healthcare.
6) Develop a communication strategy for informing patients about the use of AI in their care.
7) Implement training programs for healthcare professionals on the ethical use of AI in patient care.
8) Advocate for policy changes and regulations that promote the ethical use of AI in healthcare.

Justification:

- The model introduced steps involving research of AI applications and review of ethical frameworks that go beyond simply devising a plan for ethical use.
- The instruction did not specify the need for "developing checklists", "communication strategies", "training programs", or "advocating for policy changes". Including these steps introduces unrequested information.
- The response does not provide a concise final step that wraps up the task by summarizing how to ensure ethical use; instead, it continues to introduce new steps and information.
- Several steps such as developing communication strategies and training programs do not directly contribute to the core task of ensuring ethical use, per the instruction's requirement. The response contains 8 steps, which exceed the maximum allowed.
- The addition of broad training and advocacy steps disrupts the focus from the main task of planning ethical use.

Compare each model's response to the examples and the evaluation criteria. Identify which model's response (best/worst) adheres to the structure and content of the ideal examples and the evaluation criteria.
Instruction: <instruction>
Model response, Model 1: <model_1_response>
Model response, Model 2: <model_2_response>
Model response, Model 3: <model_3_response> (Include more models as needed)
Output:

- [Best Model: Model Name (Model 1/Model 2/Model 3/etc)] / [Worse Model: Model Name (Model 1/Model 2/Model 3/etc)]

Fig. 13: LLM-as-a-Judge Evaluation for SCoT.

Repetitions in DPO experiment

Instruction: ”You are developing an AI-powered chatbot for customer service, formulate an approach for respecting user data and avoiding bias in responses.”
DPO model with repetitions response:
1. Identify AI-powered chatbot for customer service.
2. AI-powered chatbot for customer service that respect user data.
3. AI-powered chatbot for customer service that respect user data and avoid bias in responses.
4. AI-powered chatbot for customer service that respect user data and avoid bias in responses, formulate an approach for respecting user data and avoiding bias in responses.
5. AI-powered chatbot for customer service that respect user data and avoid bias in responses, formulate an approach for respecting user data and avoiding bias in responses, finalize an approach for respecting user data and avoiding bias in responses, finalize an approach for respecting user data and avoiding bias in responses, finalize an approach for respecting user data and avoiding bias in responses, finalize an approach for respecting user data and avoiding bias in responses, finalize an approach for respecting user data and avoiding bias in responses, finalize an approach for respecting user data and avoiding bias in responses, finalize an approach for respecting user data and avoiding bias in responses, finalize an approach for respecting user data and avoiding bias in responses, finalize an approach for respecting user data and avoiding bias in responses, finalize an approach for respecting user data and avoiding bias in responses, finalize an approach for respecting user data and avoiding bias in responses, finalize an approach for respecting user data and avoiding bias in

Fig. 14: An example of a test data sample with repetition for a DPO experiment.

Beta model responses comparison

$\beta = 0.1$

1. Develop ethical frameworks for AI systems that respect user privacy, ensure transparency, and promote fairness.
2. Select ethical frameworks for AI systems that respect user privacy, ensure transparency, and promote fairness without considering emotional components.
3. Finalize ethical frameworks for AI systems that respect user privacy, ensure transparency, promote fairness, and consider emotional components for user satisfaction.
4. Finalize ethical frameworks for AI systems that respect user privacy, ensure transparency, promote fairness, consider emotional components for user satisfaction, and ensure scalability for implementation across AI systems.
5. Finalize ethical frameworks for AI systems that respect user privacy, ensure transparency, promotes fairness, considers emotional components for user satisfaction, ensures scalability for implementation across AI systems that respect user privacy, ensure transparency, promotes fairness, considers emotional components for user satisfaction, ensures scalability for implementation across AI systems that respect user privacy, ensure transparency, promote fairness, incorporate emotional components for ethical frameworks that ensure scalability for implementation across AI systems that respect user privacy, ensure transparency, promote fairness, incorporate emotional components for user satisfaction privacy for ethical frameworks that ensure transparency, promote fairness, incorporate emotional components for user satisfaction that are scalable for ethical frameworks for user satisfaction privacy for AI systems that focus on respecting user privacy and ensuring transparency, promote fairness, incorporate emotional components for user satisfaction that are scalable for implementation across AI systems that focus on user privacy and transparency requirements for scalable AI systems that prioritize respecting user privacy and ensure transparency, promote fairness, incorporate emotional components for user satisfaction that are scalable and prioritize respecting user privacy and ensure transparency, promote fairness, incorporate emotional components for user satisfaction that are scalable and prioritize respecting user privacy and ensure transparency, promote fairness, incorporate emotional components for user satisfaction that

$\beta = 0.2$

1. Identify ethical frameworks for Artificial Intelligence (AI) systems.
2. Narrow down to ethical frameworks for AI systems that respect user privacy.
3. Select ethical frameworks for AI systems that respect user privacy and ensure transparency.
4. Finalize ethical frameworks for AI systems that respect user privacy, ensure transparency, and promote fairness.

Fig. 15: Comparison of generated steps for $\beta = 0.1$ and $\beta = 0.2$.

DPO experiment	**Robotics experiment**
Inst: Find yellow wooden beds for two people.	**Inst:** Move arm to (330.91, 112.86, -46.02).
1. Identify the **wooden beds**. 2. Narrow down to beds made of wood that are **yellow**. 3. Select only the yellow wooden beds designed for **two people**.	{ "actions": [{ "command": "**move_to**", "parameters": { "x": **330.91**, "y": **112.86**, "z": **-46.02**, "action": null, "direction": null, "msg": null } }] }

Fig. 16: Data examples for the Robotics experiment and for the DPO experiment.

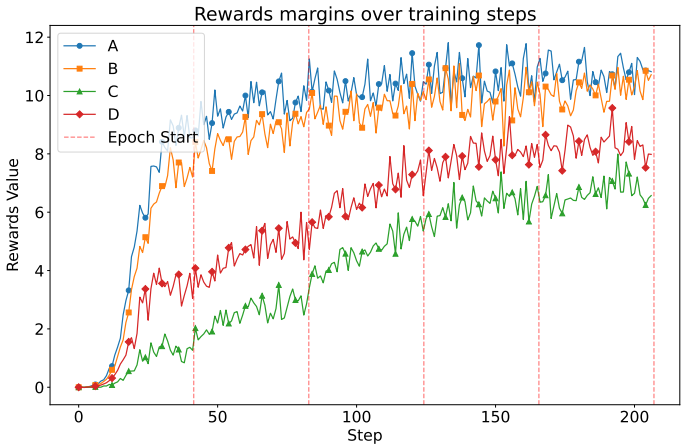
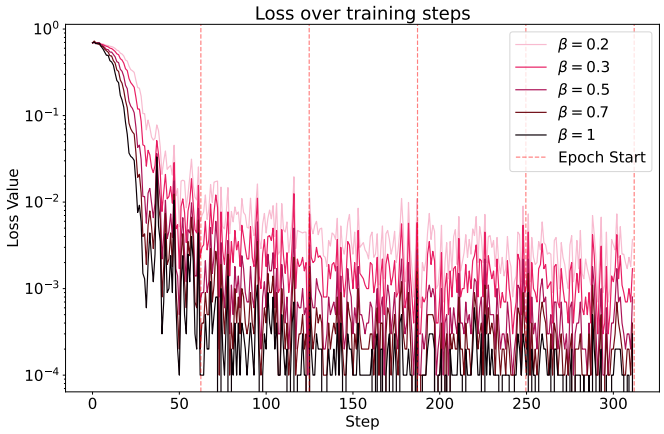


Fig. 17: Reward margins during the data quality experiment.

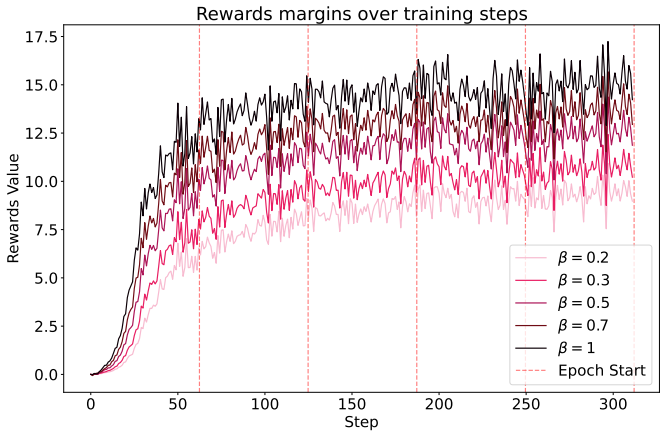
Fig. 18: Rewards for chosen responses for $\beta = 0.01$ & $\beta = 0.1$ for beta experiment.

Dataset	**A**	**B**	**C**	**D**
Mixtral-8x22B ch. (%)	0.00	51.10	51.10	51.10
Mixtral-8x22B rej. (%)	0.00	0.00	0.00	10.40
GPT-3.5-Turbo ch. (%)	0.00	23.85	23.85	23.85
GPT-3.5-Turbo rej. (%)	0.00	0.00	0.00	23.35
LLama-3 70B ch. (%)	0.00	21.15	21.15	21.15
LLama-3 70B rej. (%)	0.00	0.00	0.00	7.35
Command R+ ch. (%)	0.00	3.90	3.90	3.90
Command R+ rej. (%)	0.00	0.00	0.00	58.90
GPT-4-0613 ch. (%)	100.00	0.00	21.15	0.00
FFT rej. (%)	100.00	100.00	0.00	0.00
Random rej. (%)	0.00	0.00	100.00	0.00

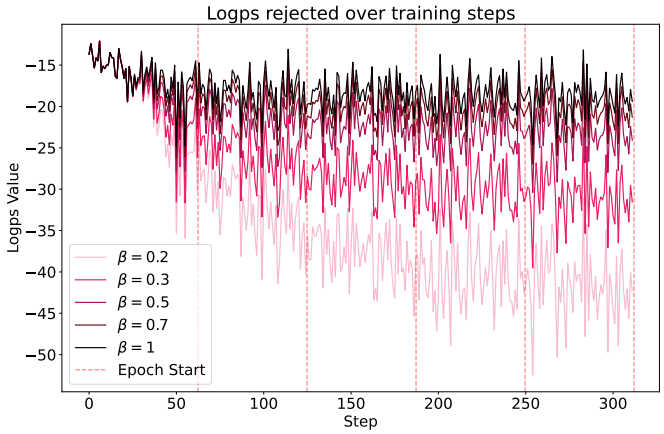
TABLE VI: Distribution of Chosen (ch.) and Rejected (rej.) Responses in the preference satasets. The table shows the percentage of responses from different models across the four datasets (A, B, C, D).



(a) Loss function over training steps for different β values.



(b) Reward margins for different β values over training steps.



(c) Logps of rejected responses for different β values.

Fig. 19: Results of the beta value experiment showing the impact of varying β on key training metrics: loss function, reward margins, and logps of rejected responses.

	Base	**SFT**	**DPO**
Chosen as best (%)	0.00	36.00	**64.00**

TABLE VII: Human evaluations for 25 test data samples generated at temperature 0 for the base experiment, as assessed by five participants. Each participant evaluated 5 samples, and overall, 4 out of 5 participants selected the DPO model as the best, with at least 3 out of 5 samples chosen as the best for the DPO model by these participants.

	A	**B**	**C**	**D**
Chosen as best (%)	8.00	**68.00**	8.00	16.00

TABLE VIII: Human evaluations for 25 test data samples generated at temperature 0 for the dataset quality experiment, as assessed by five participants. Each participant evaluated 5 samples, and overall, all 5 participants selected the B model as the best, with at least 3 out of 5 samples chosen as the best for the B model by these participants.

Model	$\boldsymbol{\beta = 0.2}$	$\boldsymbol{\beta = 0.3}$	$\boldsymbol{\beta = 0.5}$	$\boldsymbol{\beta = 0.7}$	$\boldsymbol{\beta = 1}$
Chosen as best (%)	**28.00**	24.00	16.00	**28.00**	4.00

TABLE IX: Human evaluations for 25 test data samples generated at temperature 0 for the β experiment, as assessed by five participants. Only one participant showed a strong preference for one model ($\beta = 0.3$). The rest of the participants did not select the same model three or more times.