International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Talk Smart, Talk Small: Crafting Domain-Specific LLMs for SME Customer Support

Inn Keat Ng[1], Tong Ming Lim*[2]

Submitted: 12/01/2025 **Revised:** 26/02/2025 **Accepted:** 15/03/2025

Abstract: This project addresses key challenges faced by commercial large language models (LLMs) in customer engagement, such as inconsistent responses, inaccuracies, hallucinations, and lack of follow-up questions. The goal was to develop a domain-specific LLM from scratch for small and medium enterprises (SMEs), capable of delivering relevant, consistent, and human-like responses. The methodology involved studying LLM architectures, preparing and expanding datasets, developing a base model, fine-tuning with larger domain-specific data, applying reinforcement learning, and evaluating model performance. The initial model, trained on 1.5 million tokens, lacked the language understanding needed for coherence. Scaling the dataset to 445 million tokens with general and domain-specific data improved training dynamics and model stability. Fine-tuning with 550 million tokens enhanced relevance, consistency, and human-likeness, outperforming parameter-efficient methods such as LoRA. Reinforcement learning using Identity Preference Optimization (IPO) yielded mixed results. The Normal IPO approach maintained training stability and preserved response quality at both sentence and response levels. However, the Checkpoint and EMA strategies showed fluctuating training behavior and declines in response-level consistency, human-likeness, and relevance, likely due to the small reinforcement learning dataset and instability from evolving reference models. Despite these challenges, the project demonstrated the feasibility of building a domain-specific LLM tailored for SME customer engagement. Future directions include expanding the reinforcement learning dataset, exploring alternative optimization strategies, and incorporating human feedback to further refine performance.

Keywords: *Artificial Intelligence, Customer Engagement, Fine-tuning, Large Language Model, Reinforcement Learning*

1. Introduction

1.1. Background

Small and Medium-sized Enterprises (SMEs) in Malaysia have grown significantly in recent years, contributing 38.2% to the national GDP in 2020 [1]. Despite their economic importance, SMEs face major challenges, particularly human resource constraints. They often struggle to recruit and retain employees due to lower salaries, limited career growth, and reduced job stability compared to larger firms, leading to high turnover rates [2]. This shortage of staff directly impacts customer engagement.

Financial constraints are another major issue. As noted by [3], SMEs generally have less access to capital than larger companies. This limits their ability to invest in tools like customer relationship management (CRM) software, which is expensive and requires expertise to operate effectively [4]. Without such tools, SMEs struggle to analyse customer behaviour and provide strategic service.

To address these challenges, SMEs are increasingly adopting commercial large language models (LLMs) to support customer service. LLMs can automate responses to common queries, reducing the need for large customer support teams and easing HR pressure. Additionally, their ability to analyze customer interactions helps compensate for the lack of CRM software, offering a cost-effective alternative for improving customer engagement.

1.2. Problem Statement

According to our collaborator, a leading software company in Malaysia that provides versatile digital solutions, there are several problems that they face when using the commercial LLM where the general-purpose generative AI chatbot has the following challenges:

1 Tunku Abdul Rahman University of Management and Technology (TARUMT), Kuala Lumpur – 53300, MALAYSIA

ORCID ID: 0009-0009-5343-8771

2 Tunku Abdul Rahman University of Management and Technology (TARUMT), Kuala Lumpur – 53300, MALAYSIA

ORCID ID: 0000-0003-1335-2999

** Corresponding Author Email: limtm@tarc.edu.my*

- Inability to provide consistent responses.
- Failure to deliver correct answers, even when relevant knowledge is available.
- Hallucination, leading to responses that combine inaccurate or unrelated information.
- Lack of follow-up questions when faced with unclear or unanswerable customer queries.

1.3. Objective

This project aims to study, build and validate a domain-specific LLM model from scratch for customer engagement for the selected SME companies that can provide relevant, consistent and human-like responses.

2. Related Works

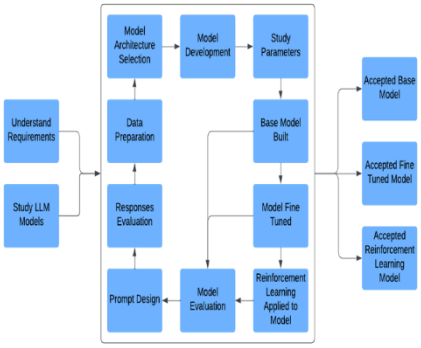

Fig. 1. Model development process workflow

The integration of large language models (LLMs) into customer service has garnered significant attention, particularly for small and medium-sized enterprises (SMEs) seeking cost-effective solutions. [5] explored the application of LLMs like GPT-4 in technical customer support, demonstrating their potential to automate tasks such as text correction, summarization, and question answering. Their findings highlighted improvements in efficiency and quality of service, while also emphasizing the necessity for quality assurance and organizational adjustments to fully leverage these technologies.

Fine-tuning pre-trained LLMs has emerged as a prevalent strategy to adapt these models for domain-specific tasks. [6] provided a comprehensive overview of methodologies for fine-tuning LLMs, outlining steps required for customizing models to specialized use cases. Their review underscored the benefits of fine-tuning in enhancing model performance for specific applications, while also discussing limitations such as potential overfitting and the need for substantial domain-specific data.

In the context of SMEs, the adoption of LLMs is often hindered by resource constraints. [7] introduced CARE, a lightweight chatbot fine-tuned using QLoRA on minimal hardware, capable of handling queries in telecommunications, medical, and banking domains. Their approach demonstrated that effective domain-specific LLMs could be developed without extensive computational resources, making them accessible to SMEs.

Despite these advancements, challenges persist in the deployment of commercial LLMs. [8] identified high development and training costs, lack of pricing transparency, and the impact of open-source alternatives as significant barriers for businesses. These factors contribute to the hesitancy among SMEs to fully adopt LLMs, highlighting the need for more affordable and transparent solutions.

Collectively, these studies illustrate the potential of LLMs to transform customer service operations, especially for SMEs. However, they also underscore the importance of addressing technical and economic challenges to facilitate broader adoption and effective implementation of these technologies.

3. Methodology

This section outlines the activities involved in designing and developing the customer engagement model. Fig. 1 illustrates the complete workflow of this process. It begins with understanding project requirements, where collaboration with stakeholders defines both the technical and functional needs of the model to ensure it is resource-efficient and suitable for real-world deployment. An in-depth study of existing large language models (LLMs) follows, analysing core components such as tokenization, embeddings, and attention mechanisms to inform architectural and design decisions. Data preparation then involves extracting, cleaning, and transforming diverse datasets into structured formats suitable for training. Based on these foundations, an appropriate model architecture is selected, considering scalability and domain-specific needs. This is followed by model development, where source code is implemented to realize the model's core functionalities. A study of model parameters is conducted to optimize performance. Using these insights, a base model is built and initially evaluated using metrics such as loss. The model is then refined through prompt design to enhance the relevance, consistency, and human-likeness of responses. Human evaluators then assess the generated outputs to identify strengths and improvement areas. Based on evaluation outcomes, the base model may undergo fine-tuning, incorporating task-specific datasets and feedback to improve performance. This results in an accepted fine-tuned model if performance standards are met. To further improve alignment with human preferences and behaviour, reinforcement learning may be applied to the fine-tuned model. This phase uses preference dataset to guide optimization, culminating in an accepted reinforcement learning model if quality benchmarks are achieved. The process concludes with the selection of the most suitable model—whether base, fine-tuned, or reinforcement-learned—balancing efficiency, accuracy, and user-centric

attributes, thereby establishing a robust foundation for future deployment.

4. Results and Discussion on the Design and Implemented Models

4.1. Base Model Architecture

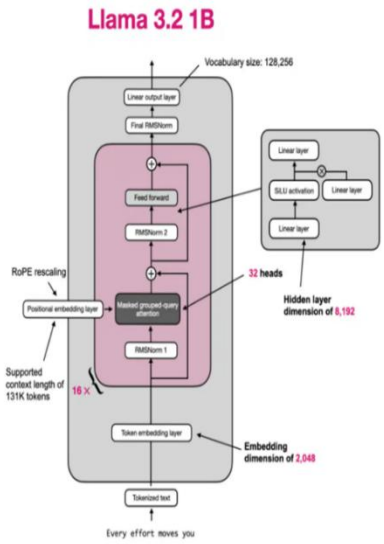

Fig. 2. LLaMA 3.2 model illustration. [9]

The LLaMA 3.2 model, proposed by [10], is built on a decoder-only transformer architecture that integrates several advanced components to enhance performance and efficiency. It utilizes a Byte Pair Encoding (BPE) tokenizer with an expanded 128,000-token vocabulary, derived from Tiktoken and enhanced to support multiple languages, enabling precise and flexible text processing. For positional encoding, it employs Rotary Position Embedding (RoPE), which captures both absolute and relative token positions to improve contextual understanding, as proposed by [11]. The model's attention mechanism adopts grouped-query attention, reducing redundancy and improving memory efficiency without compromising output quality, as introduced by [12]. RMSNorm, proposed by [13], is used for layer normalization, offering a computationally lighter alternative to traditional methods by normalizing inputs based on root mean square values. Additionally, the feed-forward network incorporates the SwiGLU activation function, which introduces a gating mechanism to capture complex, non-linear patterns, as proposed by [14]. These components—coupled with residual connections for stable gradient flow—enable LLaMA 3.2 to deliver high-quality results in tasks requiring deep language comprehension while remaining computationally efficient and scalable.

4.2. Base Model Data Preparation

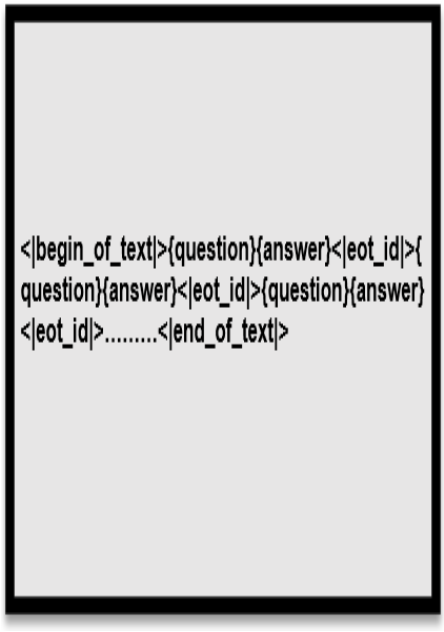

Fig. 3. Conversational data format for base model

The dataset comprises both general and domain-specific conversational data. The general portion includes large-scale multi-turn dialogues from the UltraChat dataset [15], which were filtered into two subsets: general-purpose conversations and customer engagement-related conversations based on relevant keyword presence. The domain-specific portion was developed from CRM books, scholarly articles, newly collected CRM magazines, property descriptions, and maintenance handbooks. These materials were collected in collaboration with our collaborator, a property management company focused on short- to medium-term rentals to better reflect the language and context of their operational domain. The collected content was then converted into multi-turn conversational format using the Ollama LLaMA 3.2 model. All dialogues follow a standardized structure, where each question-answer pair is separated by a <|eot_id|> token, and the entire multi-turn interaction is wrapped with <|begin_of_text|> and <|end_of_text|> tokens as depicted in Fig. 3. Due to their high initial quality, no further cleaning was necessary. This dataset was stored in a SQLite database, with a total size of approximately 2.5 GB and an estimated 450 million tokens.

4.3. Base Model Training and Parameters Setting

Table 1. Base model parameters

Parameters	Smaller Configuration	Default Configuration
Vocabulary Size	128,256	128,256
Context Length	1,024	131,072
Token Embedding Dimension	512	2,048
Number of Attention Heads per Transformer Layer	16	32
Number of Transformer Layers (Model Depth)	4	16
Hidden Dimension of the Feedforward Neural Network	4,096	8,192
Number of Key-Value Groups in Grouped-Query Attention	8	8
Base Scaling Factor for Rotary Positional Embedding (RoPE)	3,906.25	500,000.0
Frequency Scaling Factor for RoPE	32.0	32.0
Low-Frequency Component Scaling Factor for RoPE	1.0	1.0
High-Frequency Component Scaling Factor for RoPE	4.0	4.0

We implement a smaller-parameter variant of the LLaMA 3.2 model, derived from the default 1B setup described in [9], with key modifications to improve training efficiency while preserving learning capacity.

The vocabulary size is kept at 128,256, identical to the default configuration. This ensures compatibility with the original tokenizer and maintains consistency in token representation and processing. The context length is reduced from 131,072 to 1,024 tokens. This significantly lowers memory consumption and speeds up training while still allowing the model to capture meaningful contextual information for most training sequences.

The token embedding dimension is decreased from 2,048 to 512. This smaller embedding size reduces the number of parameters and computational complexity, making the model more suitable for smaller-scale training without drastically affecting its ability to learn semantic relationships. The number of attention heads per transformer layer is reduced from 32 to 16. While this simplifies the attention mechanism, it still provides sufficient diversity in attention perspectives to model contextual dependencies effectively.

The number of transformer layers, or the model's depth, is scaled down from 16 to 4. This reduction lowers training time and resource requirements while retaining enough capacity to model fundamental language structures. The hidden dimension of the feedforward neural network is halved from 8,192 to 4,096. This maintains a balance between model expressiveness and computational efficiency within each transformer block.

The number of key-value groups used in grouped-query attention is kept at 8 in both configurations. This consistency ensures that the behaviour of the attention mechanism remains unchanged and stable. The RoPE (Rotary Positional Embedding) base scaling factor is decreased from 500,000.0 to 3,906.25 to better align with the shorter context length. This adjustment ensures more precise positional encoding across shorter sequences.

The RoPE frequency scaling factor remains at 32.0, preserving the original frequency encoding behaviour and ensuring consistency across configurations. Similarly, the low-frequency and high-frequency component scaling factors for RoPE are retained at 1.0 and 4.0, respectively. These values help maintain effective encoding of both long- and short-range positional dependencies, even within the smaller context window.

Overall, these modifications create a lightweight and efficient version of the LLaMA 3.2 model that is well-suited for experimentation and resource-constrained environments, without significantly compromising its ability to model language effectively.

Table 2. Base model training parameters

Parameters	Value
Epoch	3
Batch Size	25
Optimizer	AdamW with learning rate 0.0005 and weight decay 0.1
Stride	512
Warmup Steps	20% of the total steps
Initial Learning Rate	0.00001
Minimum Learning Rate	0.00001

The base model was trained for 3 epochs using a batch size of 25. Training employed the AdamW optimizer with a maximum learning rate of 0.0005 and a weight decay of 0.1 to promote generalization. The learning rate schedule followed a linear warmup, starting from an initial learning rate of 0.00001 and increasing to its peak over the first 20% of total training steps. Following warmup, the learning rate decayed linearly to a minimum of 0.00001. A stride size of 512 was used to handle overlapping context windows efficiently during training.

4.4. Base Model Training Evaluation

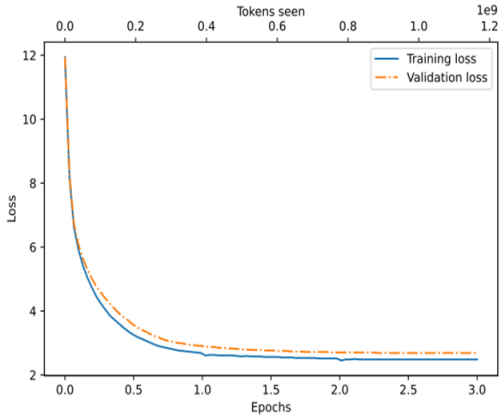

Fig. 4. Loss result from base model

As shown in Fig. 4, both training and validation losses begin at approximately 12 at the start of the first epoch and drop sharply, indicating rapid initial learning. By the end of the first epoch, the losses stabilize around 2.5 and maintain this level throughout the remaining two epochs. This steady trend reflects a smooth and consistent training process. The close alignment between training and validation losses suggests strong generalization to unseen data, with no signs of overfitting—indicating a well-trained and effectively regularized model.

4.5. Base Model Response Evaluation

For response evaluation, no prompting techniques were applied to the base model. Evaluation focused on three primary criteria—human-likeness, consistency, and relevance—assessed at both the sentence and response levels.

- Human-likeness evaluates fluency and naturalness within sentences and the coherence of full responses.
- Consistency considers the logical structure, tone stability, and reliability of responses when identical queries are presented.
- Relevance measures how directly and appropriately the content addresses the user's query while maintaining contextual alignment.

The base model successfully meets all three evaluation criteria at both the sentence and response level. It generates responses that are fluent, polite, and natural-sounding, closely resembling human communication. The tone remains consistently helpful and professional, aligned with the system prompt. In terms of consistency, the base model maintains a coherent tone throughout each response and avoids contradicting earlier parts of the conversation. It also demonstrates a clear understanding of the user's intent and provides answers that align well with the query. Regarding relevance, the model produces contextually appropriate responses that directly address the user's questions without drifting off-topic or introducing unrelated information. The answers are concise yet informative, showcasing a strong grasp of the input prompt and overall conversation flow.

4.6. Fine-tuning Design and Methods

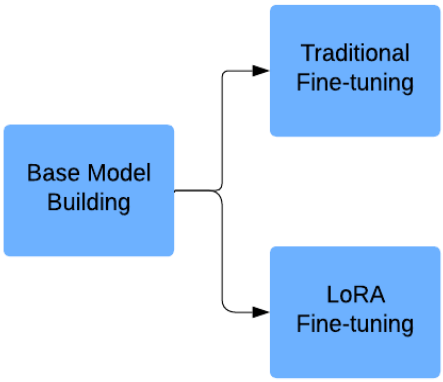

Fig. 5. Overview of the fine-tuned model training pipeline

As illustrated in Fig. 5, the base model obtained in Section 4.4 was further fine-tuned using two distinct methods: traditional fine-tuning and LoRA (Low-Rank Adaptation) fine-tuning. These approaches were implemented and evaluated to compare their performance, efficiency, and suitability for the target task.

Traditional fine-tuning involves updating all parameters of the pretrained model to adapt it to a specific downstream task. The entire weight matrix is trainable, allowing the model to compute full-rank updates through standard backpropagation. This provides high flexibility and strong in-distribution performance, especially when the target task differs significantly from the original pretraining objective or when sufficient computational resources are available [16] [17]. However, this approach is computationally and memory intensive, which can be a limitation in resource-constrained environments. Additionally, it may lead to overfitting on small datasets and can cause the model to lose generalization if fine-tuned too aggressively [18].

LoRA fine-tuning, in contrast, offers a parameter-efficient alternative by inserting two small, trainable low-rank matrices into the model while keeping the pretrained weights frozen. These matrices perform a low-rank decomposition that adjusts the model's output without altering its core parameters. LoRA significantly reduces the number of trainable parameters, enabling efficient fine-tuning even on large models with limited hardware [17]. It helps preserve the integrity of pretrained features, reducing the risk of catastrophic forgetting and improving generalization [19]. Nevertheless, LoRA's performance is sensitive to the choice of the rank parameter, and it may struggle to adapt to complex tasks that require extensive parameter reconfiguration [16] [17].

4.7. Fine-tuned Model Data Preparation

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>

{system prompt}<|eot_id|>

<|start_header_id|>{role}<|end_header_id|>

{question}<|eot_id|>

<|start_header_id|>assistant<|end_header_id|>

{response}<|eot_id|><|end_of_text|>
```

Fig. 6. Conversational data format for fine-tuned model

The dataset used for fine-tuning was constructed through a careful process of curation, augmentation, and refinement to ensure quality and alignment with the model's intended use. It integrates three high-quality sources: (1) ChatGPT-generated Q&A samples and (2) the Bitext Customer Service Tagged Training Dataset [20]—both of which are domain-specific and tailored to customer service—and (3) the LaMini-instruction dataset [21], which provides general-purpose instruction-following data.

All entries were reformatted into a unified single-turn conversational structure, as illustrated in Fig. 6. Each sample includes a system prompt, a user query, and a corresponding assistant response. To guide model behaviour, different system prompts were applied based on the data source. For LaMini entries, the following general-purpose prompt was used:

"You are a thoughtful assistant. For every task, think step-by-step before answering. Identify the goal, plan the steps, explain your reasoning briefly, then give the answer. Always prioritize clarity and logical thinking. If something is unclear, ask for clarification first."

For both ChatGPT-generated and Bitext entries, a customer service–oriented prompt was applied:

"You are a helpful and professional customer service agent for a property rental company. Your job is to answer client inquiries politely, clearly, and in a friendly, human-like manner."

In all samples, the role was standardized as "user" to ensure structural consistency across datasets. The query is treated as the user's question, and the answer is treated as the response from the assistant. The final formatted dataset was stored in an SQLite database of approximately 2.9 GB, totalling around 550 million tokens.

4.8. Fine-tuned Model Training and Parameters Setting

For traditional fine-tuning, the model parameters used are identical to those of the smaller base model configuration described in Section 4.3, as the fine-tuning process builds directly on top of the pretrained base model without modifying its architecture. Likewise, the training parameters remain consistent with those outlined in Section 4.3. This consistency ensures that any observed improvements in performance can be attributed to the updated training data rather than changes in the training procedure.

For LoRA fine-tuning, the model configuration largely mirrors that of the smaller base model in Section 4.3, with the addition of one LoRA-specific parameter: rank. This parameter determines the dimensionality of the low-rank adaptation matrices inserted into each attention layer. A higher rank enables the model to learn more task-specific patterns but also increases the number of trainable parameters. In our experiments, we explored rank values of 8 and 16 to evaluate the trade-offs between model performance and computational efficiency. All other training parameters were kept consistent with Section 4.3 to ensure a fair and controlled comparison across different fine-tuning strategies.

4.9. Fine-tuned Model Training Evaluation

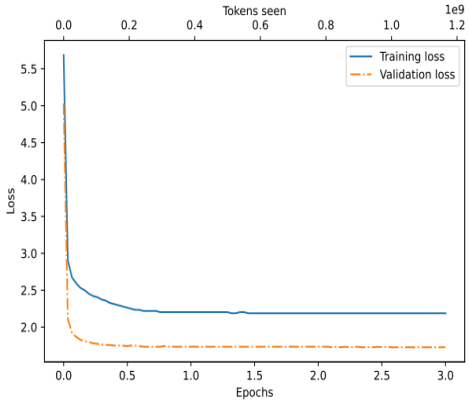

Fig. 7. Loss result from traditional fine-tuned model

As illustrated in Fig. 7, both training and validation losses start at approximately 6.0 at the beginning of the first epoch. They exhibit a sharp decline early on, indicating rapid learning progress. By the midpoint of the first epoch, the losses stabilize around 1.4 and maintain this level consistently through to the end of the third epoch. This steady trend reflects a stable and effective training process. The close alignment between the training and validation losses suggests strong generalization to unseen data, with no signs of overfitting—characteristic of a well-optimized and robust model.

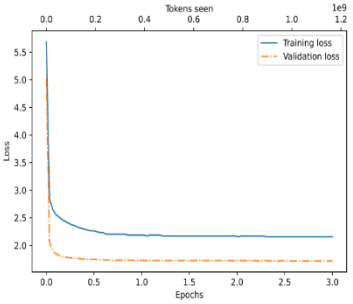

Fig. 8. Loss result from LoRA fine-tuned model of rank 8

In Fig. 8, the loss trajectory for the LoRA fine-tuned model with rank 8 begins at approximately 5.5 for both training and validation. The losses decrease rapidly during the initial training phase, eventually stabilizing at around 2.5 (training) and 1.5 (validation). Although the losses remain stable throughout the training, the consistent gap between the two curves indicates mild overfitting. Despite slightly higher final loss values compared to traditional fine-tuning, the convergence remains steady, demonstrating that the LoRA method at rank 8 enables effective and efficient adaptation, with some overfitting.

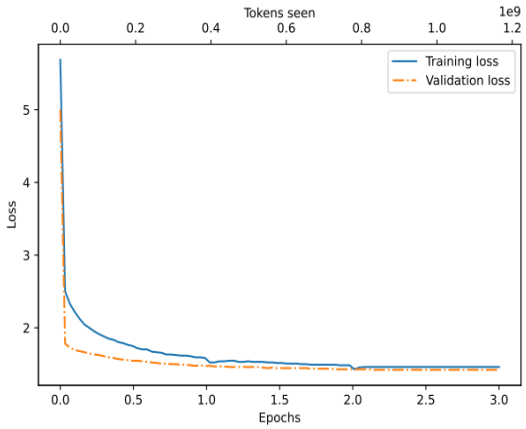

Fig. 9. Loss result from LoRA fine-tuned model of rank 16

Fig. 9 presents the loss curves for the LoRA fine-tuning configuration with rank 16. The training and validation losses, again starting around 5.5, drop sharply in the early stages and level off at approximately 2.5 and 1.5, respectively. This stable progression continues throughout the training process. Similar to the rank 8 setup, the persistent gap between the losses reflects slight overfitting. Nonetheless, the consistent convergence suggests that the rank 16 LoRA configuration remains capable of delivering successful fine-tuning outcomes, albeit with minor overfitting effects.

Overall, the evaluation highlights a clear trade-off between traditional and LoRA-based fine-tuning methods. Traditional fine-tuning achieves the best loss performance, converging to approximately 1.4 for both training and validation, indicating strong generalization and full model optimization. In contrast, LoRA-based fine-tuning—while exhibiting slightly higher final losses around 2.5 (training) and 1.5 (validation)—still shows stable convergence and efficient adaptation with significantly fewer trainable parameters.

Notably, the comparable results between LoRA rank 8 and rank 16 suggest diminishing returns from increasing rank for this specific task and dataset. These findings reinforce LoRA's effectiveness as a lightweight fine-tuning strategy, offering a balance between computational efficiency and performance, particularly when full model tuning is not feasible.

4.10. Fine-tuned Model Response Evaluation

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>
{system prompt}<|eot_id|>
<|start_header_id|>{role}<|end_header_id|>
{question}<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>
```

Fig. 10. Prompt template for fine-tuned model

For response evaluation, Fig. 10 illustrates the prompt template used during both traditional and LoRA fine-tuning. This format mirrors the instruction-style QnA structure introduced in Fig. 6, but excludes the response field, as the model is expected to generate it. The prompt begins with the system role, which is assigned the following instruction:

"You are a helpful and professional customer service agent for a property rental company. Your job is to answer client inquiries politely, clearly, and in a friendly, human-like manner."

Next, the user's input is included. For consistency with the training setup described in Section 4.6, all user roles are unified under a single identifier, "user". The assistant role follows, with the response left blank to allow the model to generate it during inference. Evaluation followed the same criteria outlined in Section 4.5: human-likeness, consistency, and relevance, assessed at both the sentence and response levels.

Table 3. Fine-tuned model response evaluation

Model	Responses
Traditional fine-tuned model	sentence-level and response-level consistency, sentence-level and response level relevance, sentence-level and response-level human-likeness
LoRA fine-tuned model (rank 8)	sentence-level human-likeness
LoRA fine-tuned model (rank 16)	sentence-level human-likeness

The traditional fine-tuned model demonstrates strong, well-rounded performance across all evaluation criteria. It consistently produces fluent, natural, and stylistically appropriate responses that align with professional customer service standards. Its outputs exhibit a clear logical flow and coherence, with high relevance at both the sentence and response levels—making it the most effective among the evaluated models.

In contrast, the LoRA fine-tuned model with rank 8 performs adequately at the sentence level, generating grammatically

correct and human-like sentences. However, it struggles to maintain global coherence and relevance throughout the full response, occasionally drifting from the query's main intent. The rank 16 LoRA model follows a similar pattern: while it generates natural-sounding individual sentences, its overall response quality is hindered by a lack of consistency and focus. These results emphasize a clear performance distinction between fine-tuning approaches. Traditional fine-tuning, with its full-parameter updates, yields the most contextually accurate, coherent, and human-like responses—effectively capturing both granular and holistic aspects of dialogue. In contrast, LoRA-based fine-tuning offers a parameter-efficient solution but falls short in achieving full response-level consistency and relevance. Interestingly, increasing the LoRA rank from 8 to 16 provides no measurable improvement in output quality, suggesting that higher rank alone does not resolve its architectural limitations. To bring LoRA-based models closer to the holistic capabilities of traditional fine-tuning, further refinement may be required.

4.11. Best Fine-tuned Model Accepted

Based on a comprehensive evaluation of both training dynamics and response quality, the traditional fine-tuned model emerges as the optimal choice. It achieves the lowest final loss (1.4) for both training and validation, reflecting superior learning efficiency compared to the LoRA variants, which converge at approximately 2.5 (training) and 1.5 (validation). More importantly, the traditional model consistently generates responses that are human-like, coherent, and contextually relevant at both the sentence and response levels. In contrast, while the LoRA models (rank 8 and 16) demonstrate parameter efficiency and stable training behaviour, their strengths are limited to sentence-level human-likeness. Their full responses lack both consistency and relevance, significantly affecting overall quality. Notably, increasing the LoRA rank from 8 to 16 offers no substantial improvement in loss reduction or output refinement, indicating diminishing returns for higher-rank configurations in this task. These findings highlight a clear trade-off: while LoRA offers resource efficiency, it comes at the expense of output quality. For applications where precision, coherence, and natural responsiveness are critical, the traditional fine-tuned model remains the most reliable and deployment-ready option.

4.12. Reinforcement Learning Design and Methods

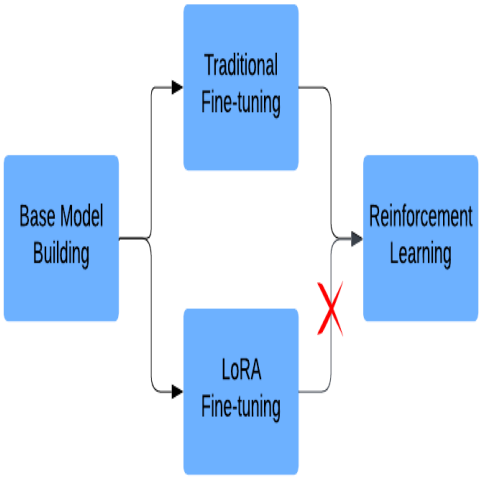

Fig. 11. Overview of the reinforcement learning pipeline

Following the development and evaluation of various fine-tuned models, the traditional fine-tuning approach was identified as the best-performing method. To further enhance this model's alignment with domain-specific human preferences, we apply reinforcement learning, specifically Identity Preference Optimization (IPO) [22]. IPO is a direct preference-based learning framework that optimizes model outputs based on comparisons between preferred (chosen) and less preferred (rejected) responses. In addition to the standard IPO setup, we integrate two reinforcement learning strategies: Exponential Moving Average (EMA) and the Checkpointed Mentor Strategy, resulting in three distinct training variants: Normal IPO, EMA IPO, and Checkpoint IPO.

Normal IPO serves as the baseline reinforcement learning approach. It directly updates the policy model by comparing the log-likelihoods of chosen and rejected responses relative to a static reference model. The training objective maximizes the preference-aligned likelihood while constraining deviations from the reference, ensuring output stability. This method avoids the need for reward models or value estimators, making it both stable and interpretable [22].

EMA IPO introduces an Exponential Moving Average mechanism [23] to improve the stability and generalization of the training process. An EMA version of the policy model is maintained throughout training by averaging the model weights over time. This smoothed model, rather than the raw policy, is typically used for inference and evaluation, offering robustness against noisy gradient updates and preventing overfitting to volatile preference samples.

Checkpoint IPO addresses the limitations of a static reference model by incorporating a Checkpointed Mentor Strategy [24]. In this method, the reference model is periodically updated with a snapshot of the current policy model. This dynamic update allows the optimization process

to evolve alongside the improving policy, avoiding both stagnation and over-regularization. As a result, the learning signal remains meaningful over time, supporting continued improvement in response quality.

These three configurations allow us to explore how enhancements in stability and adaptivity influence the effectiveness of preference-based reinforcement learning in a domain-specific customer service setting.

4.13. Reinforcement Learning Model Data Preparation

To enable Identity Preference Optimization (IPO) training, we constructed a preference-based dataset composed of question–answer (Q&A) samples, each featuring a chosen and rejected response. The initial dataset was derived by paraphrasing GPT-generated Q&As from the instruction fine-tuning dataset, as described in Section 4.7. These Q&As were reformulated and evaluated using the Ollama model, which generated alternative answers for each prompt.

For each prompt-response pair, if Ollama's response was deemed preferable to the original, it was labelled as chosen, and the original was labelled as rejected—and vice versa. This process produced approximately 20,000 Q&A samples.

To further enhance the dataset and diversify domain coverage, we generated an additional 10,000 Q&A pairs using Ollama. These were contextually grounded in the domain of our collaborators, specifically covering three user roles: tenant, owner, and prospect. Each generated conversation included clearly marked preferred (chosen) and non-preferred (rejected) answers.

All preference data were structured using a prompt-response format inspired by Figure 6, which includes a fixed system prompt:

"You are a helpful and professional customer service agent for a property rental company. Your job is to answer client inquiries politely, clearly, and in a friendly, human-like manner."

In contrast to earlier instruction-tuning datasets where the role was always "user," this dataset includes dynamic role assignments based on the specific conversational context (tenant, owner, or prospect). For every Q&A pair, two prompts were prepared—one with the chosen response and one with the rejected response—to enable direct preference learning during IPO training. The full dataset was stored in an SQLite database, totalling approximately 50MB and comprising around 10 million tokens.

4.14. Reinforcement Learning Model Training and Parameters Setting

The reinforcement learning process builds directly on the traditionally fine-tuned model without altering its architecture. Therefore, the model configuration remains consistent with the smaller base model setup previously described in Section 4.3. However, the training parameters used during the reinforcement learning phase are specifically tailored to suit the preference optimization framework and ensure stable convergence.

Table 4. Reinforcement learning model training parameters

Parameters	Value
Epoch	3
Batch Size	20
Optimizer	AdamW with learning rate 0.00001 and weight decay 0.01
Warmup Steps	20% of the total steps
Initial Learning Rate	0.000001
Minimum Learning Rate	0.000001
Update Interval Percentage	0.3
EMA rate	0.99
Beta	0.5

Training was conducted for 3 epochs with a batch size of 20, ensuring sufficient iterations over the preference dataset without overfitting. The model was optimized using the AdamW optimizer, combining adaptive learning with L2 regularization via a weight decay of 0.01, which helps maintain generalization. The learning rate was set to 1e-5 (0.00001) for the main updates, while both the initial and minimum learning rates were fixed at 1e-6 (0.000001) to avoid vanishing gradients and support gradual convergence.

A warmup phase was introduced over the first 20% of training steps, during which the learning rate increased linearly, allowing the optimizer to stabilize before reaching full-scale updates. This strategy helps reduce training volatility in early iterations, particularly when learning from noisy preference signals.

For the EMA IPO variant, an Exponential Moving Average (EMA) rate of 0.99 was used to maintain a smoothed version of the policy weights. This averaged policy helps reduce variance from noisy updates and improves the stability of model predictions during inference.

For the Checkpoint IPO variant, we employed an update interval percentage of 0.3, meaning the reference model is synchronized with the current policy every 30% of an epoch. This parameter controls how closely the reference model tracks the evolving policy.

Finally, a β (beta) parameter of 0.5 was used in the IPO loss formulation to balance the strength of the preference signal and the KL-divergence constraint, ensuring the updated policy remains aligned with the reference model while favouring preferred responses.

These parameter configurations collectively enabled a stable, effective reinforcement learning process across the different IPO strategies.

4.15. Reinforcement Learning Model Training Evaluation

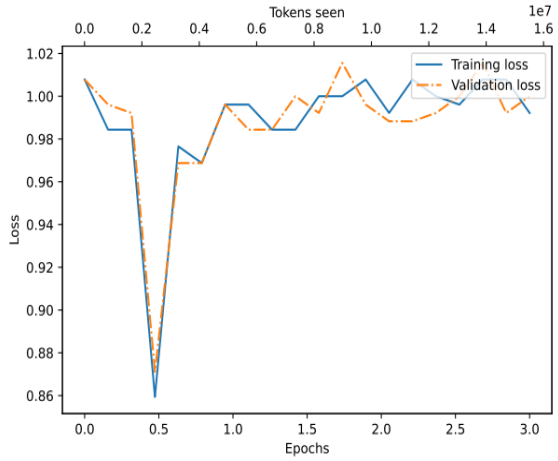

Fig. 12. Loss result from model with Normal IPO

Fig. 12 shows the training and validation loss trends for the model trained with Normal IPO. Both losses decreased rapidly within the first half of the first epoch, dropping from approximately 1.0 to 0.7. After this initial decline, the losses stabilized around 0.7 and remained consistent through the end of the third epoch. There was no indication of overfitting, as the training and validation losses closely tracked each other throughout the training process.

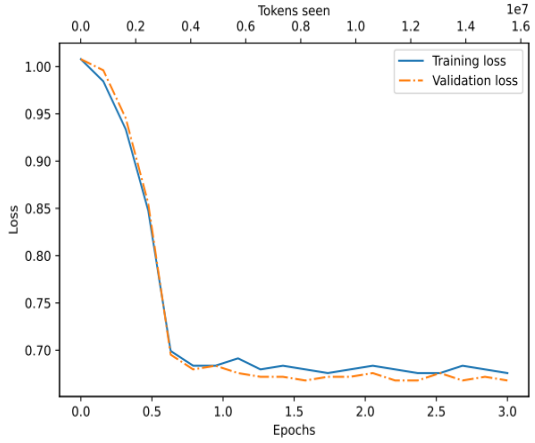

Fig. 13. Loss result from model with Checkpoint IPO

Fig. 13 illustrates the loss progression during training with the Checkpoint IPO variant. Both training and validation losses initially dropped sharply in the early stages of the first epoch, falling from around 1.0 to approximately 0.86. However, after this decline, the losses increased back to around 0.96 and fluctuated slightly around 1.0 for the remainder of the training. Despite these fluctuations, there was no sign of overfitting, as the training and validation losses remained closely aligned throughout the three epochs.

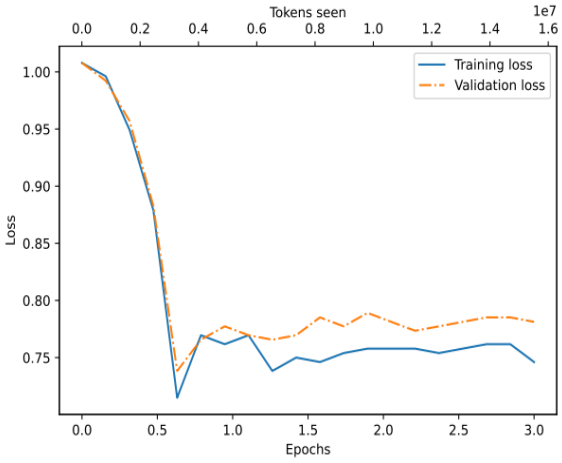

Fig. 14. Loss result from model with EMA IPO

Fig. 14 presents the loss curves for the model trained using the EMA IPO approach. The training and validation losses initially dropped from around 1.0 to approximately 0.7 during the first half of the first epoch. Following this initial improvement, the losses stabilized and fluctuated slightly around 0.75 for the remainder of the training. No signs of overfitting were observed, as the training and validation losses closely mirrored each other across all epochs.

Across the three reinforcement learning strategies evaluated, all models demonstrated effective learning without evidence of overfitting, as training and validation losses remained closely aligned throughout training. The model trained with normal IPO showed the best performance, with a rapid initial drop in loss followed by smooth and stable convergence, indicating consistent and reliable optimization. The model trained with the Checkpointed Mentor Strategy exhibited early improvements but suffered from a rise and persistent fluctuations in loss, suggesting less stable training dynamics. Meanwhile, the model trained with Exponential Moving Average (EMA) achieved early convergence but showed slightly more variability compared to the normal IPO. Based on these observations, while all approaches are viable for IPO-based reinforcement learning, the normal IPO strategy achieved the most stable and reliable training performance overall.

4.16. Reinforcement Learning Model Response Evaluation

For response evaluation, the prompt format remained consistent with the structure described in Section 4.10, with one key modification: instead of a fixed role label "user," the role field was dynamically assigned based on the contextual identity of the speaker—either tenant, owner, or prospect. This adjustment better reflects the diversity of real-world interactions and allows the model to tailor its responses more effectively to role-specific expectations. Evaluation was conducted using the criteria established in Section 4.5,

assessing human-likeness, consistency, and relevance at both the sentence and response levels.

Table 5. Reinforcement learning model response evaluation

Model	Responses
Model with Normal IPO	sentence-level and response-level consistency, sentence-level and response-level human-likeness, sentence-level and response level relevance
Model with Checkpoint IPO	sentence-level consistency, sentence-level and response-level human-likeness, sentence-level relevance
Model with EMA IPO	sentence-level and response-level consistency, sentence-level human-likeness, sentence-level relevance

The model trained with Normal IPO exhibits the strongest overall performance. It successfully mirrors the quality benchmarks established by the traditional fine-tuned model. Responses generated under this strategy are consistently human-like, contextually appropriate, and coherent at both sentence and full-response levels. The model demonstrates strong logical progression, role-awareness, and customer service tone, fulfilling the criteria of consistency and relevance across granular and holistic scopes. This makes it the most robust reinforcement learning configuration evaluated.

The Checkpoint IPO model delivers mixed results. It retains sentence-level consistency and demonstrates strong human-likeness at both the sentence and response levels. However, its performance begins to falter when considering full-response structure and topical coherence. While individual sentences may sound natural and professional, the model sometimes fails to maintain alignment with the user's query across the entire response, leading to a noticeable drop in response-level relevance and consistency. This suggests instability in the reward signal due to less frequent reference model updates.

The model using EMA IPO shows moderate success. It preserves sentence-level and response-level consistency, suggesting a stable structural output. Additionally, it maintains sentence-level human-likeness and relevance. However, it underperforms in achieving full response-level human-likeness and relevance, where responses occasionally veer off-topic or lack emotional nuance. This indicates that while EMA provides smoother guidance than the checkpoint strategy, it may dampen the distinctiveness of preferred behaviours in the reward signal.

In conclusion, among the three reinforcement learning strategies tested, Standard IPO emerges as the most effective at maintaining the response quality achieved during fine-tuning. It successfully carries forward the sentence- and response-level integrity necessary for high-quality customer service dialogue. In contrast, both Checkpoint IPO and EMA IPO reveal trade-offs between stability and adaptation, failing to fully preserve global response-level qualities. These findings highlight that while reinforcement learning adds potential for further customization and preference alignment, careful selection and tuning of the reward strategy is crucial to avoid degradation in output quality.

4.17. Best Reinforcement Learning Model Accepted

Based on a comprehensive assessment of both training dynamics and response quality, the model trained with Normal IPO emerges as the most effective and reliable reinforcement learning configuration. It demonstrated a smooth and stable loss trajectory, with a rapid initial drop followed by consistent convergence, and maintained strong performance across all evaluation metrics—achieving sentence-level and response-level consistency, human-likeness, and relevance comparable to the original fine-tuned baseline. In contrast, the Checkpoint IPO model exhibited unstable training behaviour, marked by early loss reduction followed by fluctuations, and struggled to preserve response-level consistency and topical relevance. The EMA IPO model showed more stable training than the checkpointed approach and retained local coherence and stylistic fluency, but still underperformed at the full-response level. These findings highlight Normal IPO as the only strategy capable of preserving the nuanced, high-quality outputs of the fine-tuned model while benefiting from reinforcement learning, making it the best-performing and most deployment-ready option among the configurations tested.

5. Conclusion

5.1. Achievement

This objective was successfully achieved. We developed a domain-specific LLM from scratch that demonstrated strong capabilities in producing coherent and engaging customer support responses. Notably, the traditionally fine-tuned model and Normal IPO reinforcement learning model stood out in terms of response quality. It consistently met all evaluation criteria—relevance, consistency, and human-likeness—at both the sentence and full-response levels, making them the most effective model in our study.

5.2. Discussion

Despite utilizing a relatively small base model and a modest pretraining dataset of only 450 million tokens, the model delivered strong foundational performance. This can likely be attributed to the conversational format of the data, which provides structural consistency and dialogue patterns. Such structured inputs may offer more efficient learning signals than traditional pretraining corpora, which are often heterogeneous in format and content.

In the instruction tuning phase, we expanded the training data significantly to approximately 550 million tokens. Here, traditional fine-tuning clearly outperformed parameter-efficient LoRA fine-tuning. While LoRA offers advantages in terms of memory and compute efficiency, our

results suggest that its lightweight adapter architecture was insufficient to fully capture the richness and diversity of the instruction dataset. Traditional fine-tuning, which updates all model parameters, allowed for deeper integration of instruction-following behavior, resulting in superior response quality across multiple evaluation dimensions.

During the reinforcement learning phase, Identity Preference Optimization (IPO) was applied using a more targeted, domain-specific dataset of around 10 million tokens. Among the three strategies tested—Normal IPO, Checkpoint IPO, and EMA IPO—only Normal IPO was able to preserve the high response quality achieved by the original fine-tuned model. It maintained stable loss convergence and consistently produced outputs that were coherent, relevant, and human-like at both the sentence and response levels.

In contrast, the Checkpoint and EMA variants struggled to maintain training stability and output quality. Both methods showed increased loss fluctuations and failed to preserve full-response coherence and topical alignment. A likely explanation lies in the combination of limited reinforcement learning data and the destabilizing effects of an evolving reference model. In the case of Checkpoint IPO, infrequent updates may have caused the reference to drift too far from the policy, while in EMA IPO, the overly tight coupling may have diluted the learning signal. These outcomes underscore the importance of carefully balancing reference stability and learning dynamics in preference-based reinforcement learning.

5.3. Future Works

One of the crucial directions for future work is to significantly increase the size of the datasets used across all stages—pretraining, fine-tuning, and reinforcement learning. A larger and more diverse dataset would improve language fluency, domain understanding, and the ability to capture subtle user preferences, ultimately leading to more capable and contextually aware models.

Another promising avenue is to explore alternative reinforcement learning strategies beyond Identity Preference Optimization (IPO). While Normal IPO demonstrated an ability to preserve response quality in this project, the Checkpoint and EMA variants highlighted potential pitfalls in reinforcement learning design. Techniques such as Reinforcement Learning with Human Feedback (RLHF), Direct Preference Optimization (DPO), or reward modelling based on human-annotated quality scores could offer more flexible and adaptive methods for improving conversational abilities. These alternatives may better balance optimization stability with the capacity to learn nuanced conversational behaviours, helping models generalize to diverse and realistic user interactions.

In addition to expanding data and revisiting learning strategies, future work should consider integrating human-in-the-loop feedback during training. Real-time or post-hoc evaluations by human annotators can provide more accurate and granular assessments of response quality, including tone, helpfulness, and user satisfaction. This feedback can be used to refine reward functions or guide policy updates in a more targeted manner, improving alignment with actual user expectations.

Lastly, applying more advanced fine-tuning techniques—such as QLoRA, AdaLoRA, or other parameter-efficient methods—could offer a middle ground between computational efficiency and training effectiveness. These methods may help retain the benefits of full fine-tuning while reducing resource demands, particularly when scaling up model sizes or incorporating continual learning pipelines.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] DOSM, "Department of Statistics Malaysia," *www.dosm.gov.my*, Jul. 28, 2021. https://www.dosm.gov.my/portal-main/release-content/small-and-medium-enterprises-smes-performance-2020

[2] M. F. Abas, P. Pardiman, and S. Supriyanto, "Unlocking Human Potential: A Literature Review on HR Challenges and Innovations in SME Entrepreneurship," Jurnal Manajemen Bisnis, vol. 11, no. 2, pp. 785–799, Jun. 2024, doi: https://doi.org/10.33096/jmb.v11i2.837.

[3] S. M. Yong, "4th Industry Revolution Digital Marketing Adoption Challenges in SMEs and Its Effect on Customer Responsiveness," Information Management and Business Review, vol. 15, no. 2(I)SI, pp. 152–172, Jun. 2023.

[4] V. Obradovic, "CRM software as a service and importance of the approach for SMEs," IJEEC - INTERNATIONAL JOURNAL OF ELECTRICAL ENGINEERING AND COMPUTING, vol. 6, no. 1, Jun. 2022, doi: https://doi.org/10.7251/ijeec2206042o.

[5] J. Wulf and J. Meierhofer, "Utilizing Large Language Models for Automating Technical Customer Support," arXiv.org, 2024. https://arxiv.org/abs/2406.01407

[6] D. M. Anisuzzaman, J. G. Malins, P. A. Friedman, and Z. I. Attia, "Fine-Tuning LLMs for Specialized Use Cases," *Mayo Clinic Proceedings: Digital Health*, vol. 3, no. 1, Nov. 2024, doi: https://doi.org/10.1016/j.mcpdig.2024.11.005.

[7] A. Dutta, N. Ghosh, and A. Chatterjee, "CARE:

A QLoRA-Fine Tuned Multi-Domain Chatbot With Fast Learning On Minimal Hardware," *arXiv.org*, 2025. https://arxiv.org/abs/2503.14136 (accessed May 23, 2025).

[8] InfoWorld, "3 big challenges of commercial LLMs," *InfoWorld*, Nov. 27, 2023. https://www.infoworld.com/article/2335381/3-big-challenges-of-commercial-llms.html

[9] rasbt, "GitHub - rasbt/LLMs-from-scratch: Implementing a ChatGPT-like LLM in PyTorch from scratch, step by step," GitHub, 2023. https://github.com/rasbt/LLMs-from-scratch

[10] A. Dubey *et al.*, "The Llama 3 Herd of Models," *arXiv.org*, 2024. https://arxiv.org/abs/2407.21783

[11] J. Su, Y. Lu, S.-F. Pan, B. Wen, and Y. Liu, "RoFormer: Enhanced Transformer with Rotary Position Embedding," Apr. 2021, doi: https://doi.org/10.48550/arxiv.2104.09864.

[12] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai, "GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints," *arXiv.org*, Oct. 23, 2023. https://arxiv.org/abs/2305.13245

[13] B. Zhang and R. Sennrich, "Root Mean Square Layer Normalization," *arXiv.org*, Oct. 16, 2019. https://arxiv.org/abs/1910.07467

[14] N. Shazeer, "GLU Variants Improve Transformer," *arXiv:2002.05202 [cs, stat]*, Feb. 2020, Available: https://arxiv.org/abs/2002.05202

[15] N. Ding, "ultrachat," *Huggingface.co*, 2025. https://huggingface.co/datasets/stingning/ultrachat (accessed May 24, 2025).

[16] R. Shuttleworth, J. Andreas, A. Torralba, and P. Sharma, "LoRA vs Full Fine-tuning: An Illusion of Equivalence," *arXiv.org*, 2024. https://arxiv.org/abs/2410.21228 (accessed May 25, 2025).

[17] E. J. Hu *et al.*, "LoRA: Low-Rank Adaptation of Large Language Models," *arXiv:2106.09685 [cs]*, Oct. 2021, Available: https://arxiv.org/abs/2106.09685

[18] A. Kumar, A. Raghunathan, R. Jones, T. Ma, and P. Liang, "Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution," *arXiv:2202.10054 [cs]*, Feb. 2022, Available: https://arxiv.org/abs/2202.10054

[19] Y. Zeng and K. Lee, "The Expressive Power of Low-Rank Adaptation," *arXiv (Cornell University)*, Oct. 2023, doi: https://doi.org/10.48550/arxiv.2310.17513.

[20] Bitext, "Bitext-customer-support-llm-chatbot-training-dataset," *Huggingface.co*, 2025. https://huggingface.co/datasets/bitext/Bitext-customer-support-llm-chatbot-training-dataset

[21] M. Wu, A. Waheed, C. Zhang, M. Abdul-Mageed, and A. F. Aji, "LaMini-LM: A Diverse Herd of Distilled Models from Large-Scale Instructions," *arXiv.org*, May 24, 2023. https://arxiv.org/abs/2304.14402

[22] M. G. Azar *et al.*, "A General Theoretical Paradigm to Understand Learning from Human Preferences," *arXiv.org*, Nov. 21, 2023. https://arxiv.org/abs/2310.12036

[23] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging Weights Leads to Wider Optima and Better Generalization," *arXiv:1803.05407 [cs, stat]*, Feb. 2019, Available: https://arxiv.org/abs/1803.05407

[24] P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *arXiv.org*, 2017. https://arxiv.org/abs/1706.03741