

# Fine-Tuning Gemma-2 with LoRA for Response Prediction Aligning with Human Preferences

Stanford CS224N Custom Project

**Bocheng Dai**

Department of Management Science and Engineering  
Stanford University  
bd367687@stanford.edu

**Tiantian Meng**

Department of Management Science and Engineering  
Stanford University  
meng612@stanford.edu

**Suye Shen**

Department of Management Science and Engineering  
Stanford University  
suyeshen@stanford.edu

## Abstract

Deploying Large Language Models (LLMs) effectively for chatbot interactions is challenging due to systematic biases—verbosity bias (favoring longer responses), position bias (favoring earlier responses), and self-enhancement bias (favoring overly confident answers). These biases often lead to counterintuitive chatbot responses misaligned with human preferences. We aim to fine-tune the Gemma-2-2B-it model [1] using Low-Rank Adaptation (LoRA) [2], PISSA initialization [3], and Test-Time Augmentation (TTA) [4] to predict human preferences in head-to-head chatbot interactions, specifically addressing these systematic biases. Our experiments on the LMSYS Chatbot Arena dataset demonstrate that our best-performing configuration (LoRA with TTA, without PISSA initialization) achieves superior accuracy (49.22%) and log loss (1.0373), outperforming baseline models (LMSYS XGBoost [5] and pretrained Gemma-2-2B-it). These findings highlight that strategically addressing systematic biases significantly improves chatbot alignment with human judgments.

## 1 Key Information to include

- Mentor: Mingjian Jiang

## 2 Introduction

Large Language Models (LLMs) have transformed conversational AI, enabling chatbots to deliver increasingly sophisticated responses. However, aligning these models closely with human preferences remains challenging due to systematic biases, notably verbosity bias (favoring lengthy answers), position bias (favoring responses based on their presented order rather than their intrinsic quality), and self-enhancement bias (favoring overly confident yet potentially inaccurate responses). These biases degrade user satisfaction, causing chatbot interactions to diverge from user expectations.

Systematic biases arise primarily from training data distributions, ranking methods, and architectural limitations, complicating accurate predictions of user preferences. Existing approaches, including traditional feature-engineered models like XGBoost and pretrained models such as Gemma, attempt to address this issue but often fail to capture the subtleties of human judgment effectively.

To overcome these challenges, we propose fine-tuning the Gemma-2-2B-it model with Low-Rank Adaptation (LoRA) [2], Principal Singular values and Singular vectors Adaptation (PISSA), and Test-Time Augmentation (TTA) [4]. LoRA allows parameter-efficient adaptation, and TTA reduces input variability, addressing biases effectively. While PISSA aims to optimize LoRA initialization through singular value decomposition for faster convergence, our experiments revealed that this initialization overly constrained specific transformer modules, limiting their flexibility during training and resulting in suboptimal performance.

We evaluated our approach on the LMSYS Chatbot Arena dataset, demonstrating that combining LoRA and TTA [4] yields the best performance, achieving 49.22% accuracy and 1.0373 log loss, outperforming pretrained Gemma-2-2B-it and XGBoost baselines. Our results highlight the importance of targeted bias mitigation and efficient fine-tuning, providing insights critical for developing robust, human-aligned conversational AI systems.

### 3 Related Work

Preference modeling in Large Language Models (LLMs) has become increasingly important to ensure chatbot responses align effectively with nuanced human judgments. Reinforcement Learning from Human Feedback (RLHF), introduced by [6], significantly improved model alignment but required extensive human annotations. The Chatbot Arena dataset proposed by [7] addresses these limitations by providing a large-scale dataset of head-to-head chatbot comparisons, allowing for scalable training and evaluation based on user feedback. However, systematic biases—such as verbosity bias (favoring longer responses), position bias (favoring earlier responses), and self-enhancement bias (favoring overly confident responses)—persistently affect preference prediction accuracy.

Traditional machine learning methods, notably XGBoost, have demonstrated success in structured feature extraction and ranking tasks. The LMSYS XGB baseline [5] provides an interpretable and efficient alternative but lacks the deeper contextual comprehension offered by LLM-based approaches. Recently, parameter-efficient fine-tuning methods such as Low-Rank Adaptation (LoRA), introduced by [2], have significantly reduced computational costs by training only low-rank adapters. Additionally, Principal Singular values and Singular vectors Adaptation (PISSA) [8] optimizes LoRA initialization via Singular Value Decomposition, although preliminary experiments indicate potential optimization challenges with PISSA in certain scenarios.

The Gemma-2 family of models, developed by Google DeepMind [1], offers an efficient, open-weight architecture specifically optimized for diverse NLP tasks. Gemma-2 leverages Grouped-Query Attention and interleaved local-global attention to enhance computational efficiency and contextual understanding, positioning itself as an ideal backbone for efficient sequence classification tasks [9]. However, evaluating Gemma-2 specifically for chatbot preference prediction, especially under systematic biases, remains underexplored.

Robust evaluation and bias mitigation strategies are essential for effective preference modeling. Test-Time Augmentation (TTA), successfully utilized in NLP contexts [10], enhances model robustness by averaging predictions across augmented inputs. Our approach integrates LoRA fine-tuning, PISSA initialization, and TTA with Gemma-2-2B-it, systematically evaluating these methods against robust baselines, including LMSYS XGB [5] and pretrained Gemma-2-2B-it [1]. This integration systematically addresses identified systematic biases and significantly advances robustness in preference prediction.

## 4 Approach

### 4.1 Task

The task involves developing and fine-tuning a classification model to predict human preference outcomes in comparative chatbot interactions. Specifically, given a user prompt and two responses generated by distinct LLMs, the model must predict which response a human judge would prefer,

or if the user is equally likely to prefer both responses. This involves addressing systematic biases inherent in the training data and model architecture.

For example, given the following input:

- **Prompt:** "What is the difference between a marriage license and a marriage certificate?"
- **Response A:** "A marriage license is a legal document that allows a couple to get married. A marriage certificate is proof that the marriage has taken place."
- **Response B:** "A marriage license and a marriage certificate serve different purposes. A license is issued before the marriage, while the certificate is issued after the ceremony."

The model is expected to output probabilistic predictions indicating the likelihood of each response being preferred:

- *winner\_model\_a*: 0.40
- *winner\_model\_b*: 0.55
- *winner\_tie*: 0.05

## 4.2 Model Architecture & Fine-tuning

Our fine-tuning process is structured around the Gemma-2-2B-it model [11], which we adapt using LoRA [2] to efficiently align with human preference data. We initially intended to use the larger Gemma-2-9B-it model, but due to VRAM limitations and memory fragmentation on our virtual machine, it was infeasible to load and train the model without exceeding hardware constraints.

The Gemma-2-2B-it architecture, depicted in Figure 1, follows a standard transformer-based structure with multi-head attention, feed-forward layers, and layer normalization, making it suitable for sequence classification tasks.

We employ Gemma2ForSequenceClassification, which is loaded from a pre-trained checkpoint (config.checkpoint) and fine-tuned with three output labels representing human preference probabilities (favoring Response A, Response B, or a tie). The model runs in bfloat16 precision for optimized computational efficiency.

To enhance fine-tuning, we integrate PISSA [3], ensuring optimal initialization of LoRA [2] weights through SVD decomposition. Additionally, we implement Test-Time Augmentation (TTA) [4], where response pairs are swapped during inference, and results are averaged to improve robustness. Instead of altering the standard LM head, we introduce a three-layer linear classifier, refining token representations for preference modeling.

For LoRA-based adaptation, we apply modifications to key transformer modules, including query (q\_proj), key (k\_proj), value (v\_proj), output (o\_proj), and gate projections (gate\_proj). The LoRA configuration [2] sets a rank of 64 (lora\_r), a scaling factor of 64 (lora\_alpha), and a dropout rate of 0.05 (lora\_dropout). To control adaptation depth, LoRA is only applied to trainable layers, skipping frozen ones based on the config.freeze\_layers setting.

To ensure robust training, we disable caching and prepare the model for k-bit quantization training, reducing memory footprint while maintaining expressiveness. The final fine-tuned model is wrapped using PEFT, making it computationally efficient for preference classification tasks. As illustrated in Figure 2, the LoRA method significantly reduces memory requirements by applying low-rank adapters to transformer layers while keeping the base model in a lower-precision format, enabling efficient fine-tuning under resource constraints.

## 4.3 Baselines

To assess the effectiveness of our fine-tuning approach, we compare our model against two baselines: LMSYS XGB Baseline [5] and Pretrained Gemma-2. The LMSYS XGB Baseline [5] is an XGBoost-based classifier trained on extracted structured features from response pairs. This serves as a strong non-neural baseline that provides insights into how well statistical methods perform in predicting preference selection. The Gemma-2 (Pretrained) baseline represents the original Gemma-2-2B-it model without any fine-tuning. By evaluating our fine-tuned model against these baselines, we

can quantify the improvements introduced by LoRA [2] adaptation, PISSA [3] initialization, and Test-Time Augmentation in aligning with human preference predictions.

#### 4.4 Training Process

The training process is implemented using the Hugging Face Trainer API with custom modifications. We initialize the Gemma-2-2B-it model and tokenize the dataset using our CustomTokenizer. The dataset undergoes efficient mapping and padding before training begins.

Our training configuration includes gradient accumulation (16 steps) to effectively handle large batch sizes while mitigating memory constraints. The optimizer used is AdamW with a linear learning rate scheduler, and gradient checkpointing is enabled to reduce memory overhead. We employ R-Drop (Regularized Dropout) training via a custom RDropTrainer, which computes Kullback-Leibler (KL) divergence loss between multiple forward passes, helping to improve generalization.

Additionally, we incorporate LoRA Exponential Moving Average (LoRA EMA) to stabilize weight updates. During training, metric-based evaluation (log loss, accuracy) is conducted periodically, ensuring proper performance tracking.

### 5 Experiments

#### 5.1 Data

We use the LMSYS - Chatbot Arena Human Preference Predictions dataset as the primary source for model training and evaluation. This dataset contains 57,485 labeled examples, each containing a user prompt, two chatbot-generated responses, and a binary label indicating human preference for one response or a tie. To create a robust evaluation framework, we perform a 80:20 train-test split on the provided data, allocating 45,988 rows for training and 11,497 rows for testing.

In addition, to address potential limitations and biases in the training dataset and enhance generalization, we generate bias-specific validation sets targeting verbosity bias, position bias and self-enhancement bias (examples illustrating these biases are shown in Figures 3–5). For each bias type, we create 2,000 examples, resulting in an additional 6,000 rows private validation dataset to test the model’s robustness and alignment with human preferences. The bias-specific dataset [12] was created as follows:

- **Verbosity Bias:** Construct scenarios where one response is more verbose than the other, assessing whether the model can predict human preference for concise yet informative answers when verbosity is a factor.
- **Position Bias:** Present responses in different orders to determine whether the model’s preferences are influenced by the position of the response.
- **Self-Enhancement Bias:** Create cases where one response is overconfident but factually incorrect, while the other is cautious yet accurate, testing if the model aligns with human preference for factual correctness over confidence.

The crafting of this dataset uses reinforcement learning from human feedback (RLHF) techniques, where LLMs generate candidate responses under predefined scenarios tailored to each bias, simulating user evaluations to reduce manual effort. To ensure quality and mitigate model-induced distortions, we use a manual review process combined with human-labeled validation.

#### 5.2 Evaluation method

We evaluate our model performance using two primary metrics: accuracy and log loss. Accuracy measures the proportion of correct predictions, indicating how effectively the model predicts human-preferred chatbot responses. Log loss, also known as cross-entropy loss, quantifies the uncertainty of predictions, effectively capturing how confidently and accurately the model assigns probabilities to the true labels.

We chose these metrics because they provide complementary insights—accuracy clearly demonstrates overall prediction correctness, while log loss offers a nuanced view of prediction confidence. Lower

log loss values indicate better calibration and reliability of the model’s predicted probabilities. We report and compare these metrics across various experimental configurations, including different combinations of Test-Time Augmentation (TTA) and PISSA initialization, alongside baselines like LMSYS XGBoost and a random model with no training.

### 5.3 Experimental details

The Gemma-2-2B-it checkpoint is used as the base model, with a maximum sequence length of 3072 tokens to accommodate extended chatbot responses. We employ a K-fold cross-validation approach ( $n\_splits = 5$ ), ensuring generalizability across different data splits. Training is optimized using AdamW (8-bit precision) with a learning rate of  $1e-4$  and warmup steps set to 20 to stabilize early learning. Our batch size is set at 2 per device, with gradient accumulation over 16 steps, a strategy that balances computational constraints with stable updates.

To enhance model efficiency, we fine-tune LoRA-specific parameters, using a rank of 64 ( $lora\_r$ ) with an adaptive scaling factor of 64 ( $lora\_alpha$ ) and dropout probability of 0.05 ( $lora\_dropout$ ). LoRA bias is disabled to prevent unnecessary parameter updates. Notably, all transformer layers remain trainable ( $freeze\_layers = 0$ ), allowing the model to fully adapt to preference-based ranking.

Due to quota limitations, we were unable to access NVIDIA A100 GPUs and instead conducted training on NVIDIA L4 48GB. This constraint necessitated switching from the Gemma-2-9B-it model to the Gemma-2-2B-it model to ensure successful model loading and training within our computational environment. Training is performed using LoRA [2] fine-tuning.

### 5.4 Results

We present the quantitative results obtained from various model configurations evaluated using the LMSYS Chatbot Arena dataset combined with our bias-specific validation set. Table 1 summarizes these results, comparing our proposed fine-tuning configurations against baseline models.

Table 1: Comparison of Different Model Configurations

Setting	Accuracy	Log Loss
Fine-Tuning with TTA, PISSA Initialization, Three-Scoring Layers	0.4726	1.0522
Fine-Tuning with No TTA, PISSA Initialization, Three-Scoring Layers	0.4341	1.0715
Fine-Tuning with TTA, No PISSA Initialization, Three-Scoring Layers	<b>0.4922</b>	<b>1.0373</b>
Fine-Tuning with TTA, PISSA Initialization, Single-Scoring Layer	0.3129	1.2574
XGBoost Benchmark (Feature Engineering)	0.4629	1.0582
No Fine-Tuning (Pretrained Gemma-2-2B-it)	0.2978	3.0237

The best-performing configuration—LoRA fine-tuning combined with TTA, but without PISSA initialization—achieved an accuracy of 49.22% and a log loss of 1.0373, surpassing both the traditional feature-engineered XGBoost model (46.29% accuracy, 1.0582 log loss) and the pretrained Gemma-2-2B-it baseline (29.78% accuracy). These outcomes align closely with our initial hypothesis, confirming that combining parameter-efficient fine-tuning (LoRA) with robustness-enhancing augmentation (TTA) effectively mitigates systematic biases inherent in chatbot response prediction tasks.

The relatively modest improvement over the XGBoost benchmark is notable. While we initially anticipated a larger performance gain from neural models, the observed margin suggests that traditional feature-based methods remain robust competitors, especially when the neural model size is limited by computational constraints (Gemma-2-2B-it). This indicates that future exploration with larger models, such as Gemma-2-9B-it, might yield more pronounced performance improvements.

The inferior performance of the configuration that incorporated PISSA initialization alongside TTA (47.26% accuracy) was somewhat unexpected. Despite theoretical advantages—namely, optimal weight initialization via singular value decomposition—our empirical findings suggest that PISSA might overly constrained certain transformer layers, hindering their flexibility during fine-tuning. This limitation emphasizes the importance of careful empirical validation of theoretical optimization methods, especially when applied to specific components of complex models.

Moreover, the substantially lower performance (31.29% accuracy, 1.2574 log loss) of the single-layer scoring architecture underscores the necessity of deeper, multi-layer classifiers in effectively transforming token representations to align with human preferences. Finally, the considerable performance gap between our fine-tuned model and the pretrained baseline (29.78% accuracy, 3.0237 log loss) highlights the critical importance of targeted adaptation techniques in achieving meaningful advancements.

Overall, the quantitative results support our initial hypothesis: strategic, targeted fine-tuning techniques, particularly LoRA and TTA, are crucial for enhancing the model’s alignment with human preferences, confirming the viability and effectiveness of our approach.

## 6 Analysis

We systematically explore the performance of various fine-tuning configurations of the Gemma-2-2B-it model and subsequently compare the effectiveness of neural versus traditional non-neural methods in predicting human preferences.

### 6.1 Different Gemma-2-2B-it Fine-Tuning Configurations

We evaluate several configurations of the Gemma-2-2B-it model to determine the impact of various fine-tuning components on model performance.

#### 6.1.1 Impact of Test-Time Augmentation (TTA)

Without TTA, the result has relatively low accuracy (0.4341) and high log loss (1.0715). The absence of TTA likely reduced model robustness, suggesting that TTA significantly contributes to handling input variability and biases.

#### 6.1.2 Impact of PISSA Initialization

Without PISSA initialization, the result achieves the highest accuracy (0.4922) and lowest log loss (1.0373), demonstrating that excluding PISSA initialization improved fine-tuning effectiveness. When employing PISSA initialization, the model’s parameters were initialized using Singular Value Decomposition (SVD) to adapt LoRA weights. In our implementation, we specifically apply PISSA to key transformer modules including query projection (q\_proj), key projection (k\_proj), value projection (v\_proj), output projection (o\_proj), and gate projection (gate\_proj). Although intended to optimize initial weights for better convergence, this targeted initialization might overly constrain these critical layers, limiting their flexibility in adjusting parameters during training[13]. Consequently, the restricted adaptability may lead to suboptimal convergence and negatively impact the model’s overall performance compared to standard initialization methods.

#### 6.1.3 Impact of Scoring Layer Architecture (Single vs. Three-Layer)

The single scoring layer configuration performed notably worse, achieving an accuracy of only 0.3129 and a log loss of 1.2574. The single-layer design directly maps hidden representations to the prediction task, which is fundamentally different from the original task of predicting the next token. This mismatch makes it challenging for the model to adapt effectively. Multiple scoring layers help alleviate this by gradually transforming representations into more suitable features for preference modeling.

#### 6.1.4 Impact of Fine-Tuning

Without any fine-tuning, the model demonstrated the worst results. Without adaptation to human preferences, the model produces essentially random outputs (accuracy 0.2978, log loss 3.0237), underscoring the critical need for targeted fine-tuning strategies.

### 6.2 Comparison between Neural and Non-Neural Models

Comparing the neural model (Gemma-2-2B-it with TTA, with PISSA, Three scoring layers) against a traditional non-neural model (XGBoost), the neural model marginally outperformed XGBoost,

with accuracy at 0.4726 versus 0.4629 and log loss of 1.0522 versus 1.0582, respectively. Although the neural model showed limited improvement, this modest enhancement might be partially due to the relatively small size of the neural model (2-bit) imposed by our hardware constraints[14]. It is plausible that using a larger neural model, such as a 9-bit variant, could further improve performance by offering greater modeling capacity. The modest improvement highlights the effectiveness of feature engineering with XGBoost but also suggests potential advantages of neural models under improved computational resources.

Neural models have inherent advantages over non-neural methods, primarily due to their ability to capture nuanced semantic relationships and contextual dependencies within sentences. Further improving neural model performance could involve more extensive fine-tuning, introducing architectures specifically designed for classification tasks (e.g., deeper classification layers), or using advanced training techniques like reinforcement learning from human feedback (RLHF)[15]. These enhancements could better leverage the neural model’s strengths and more effectively capture semantic nuances critical to accurately predicting human preferences.

Overall, while the neural model demonstrates promising performance, the relatively small improvement underscores the challenge of surpassing traditional methods without sufficient model size and appropriate architectural modifications, highlighting the importance of further optimization and targeted fine-tuning strategies.

## 7 Conclusion

Our work investigated fine-tuning strategies for the Gemma-2-2B-it model to improve its performance in predicting human preferences in chatbot interactions. Specifically, we explored Low-Rank Adaptation (LoRA), Test-Time Augmentation (TTA), PISSA initialization, and variations in scoring layer architecture. To address systematic biases in existing data, we developed a custom dataset to enhance the model’s generalization capabilities.

Experimental results using our custom dataset combined with the LMSYS Chatbot Arena dataset indicated that the optimal configuration utilized LoRA and TTA without PISSA initialization, achieving the highest accuracy (49.22%) and lowest log loss (1.0373). While TTA significantly improved robustness, PISSA initialization constrained key transformer layers, negatively impacting performance. Moreover, adopting a multi-layer scoring architecture rather than a single layer greatly enhanced adaptability to the preference prediction task.

Our experiments were limited by computational resources, restricting us to the relatively small Gemma-2-2B-it model. Future work could extend this analysis by using larger models such as Gemma-2-9B-it, experimenting with improved LoRA initialization methods, and incorporating advanced training approaches like reinforcement learning from human feedback.

### Team contributions (Required for multi-person team)

- Bocheng Dai: Primarily responsible for building the model architecture, fine-tuning setup, parameter tuning, and contributing to report writing.
- Suye Shen: Mainly handled the setup of the XGBoost benchmark and contributed to report writing.
- Tiantian Meng: Primarily responsible for model evaluation and data collection, also actively participating in report writing.

Overall, all team members contributed equally to the project’s workload.

## References

- [1] Google AI. Gemma: Open weight efficient models, 2024.
- [2] Edward J. Hu, Yelong Shen, Phillip Wallis, Zhi Hern Loh, Vikram N. Rangamani, Raghavendran G. Swaminathan, Christopher Ré, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.
- [3] Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models, 2024.
- [4] Ildoo Kim, Younghoon Kim, and Sungwoong Kim. Learning loss for test-time augmentation. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4163–4174. Curran Associates, Inc., 2020.
- [5] Sercan Yesiloz. Lmsys xgb baseline, 2024. Accessed: 2024-02-08.
- [6] Long Ouyang, Jeffrey Wu, Xi Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, and Alex Ray. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.
- [7] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, and Siyuan Hao. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.
- [8] Meng et al. Principal singular values and singular vectors adaptation (pissa), 2024.
- [9] Riviere et al. Grouped-query attention and interleaved local-global attention, 2024.
- [10] Kim et al. Test-time augmentation for nlp, 2020.
- [11] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshtir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis



- Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024.
- [12] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc., 2023.
- [13] Shaowen Wang, Linxi Yu, and Jian Li. Lora-ga: Low-rank adaptation with gradient approximation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 54905–54931. Curran Associates, Inc., 2024.
- [14] Washington Cunha, Vitor Mangaravite, Christian Gomes, Sérgio Canuto, Elaine Resende, Cecilia Nascimento, Felipe Viegas, Celso França, Wellington Santos Martins, Jussara M. Almeida, Thiersen Rosa, Leonardo Rocha, and Marcos André Gonçalves. On the cost-effectiveness of neural and non-neural approaches and representations for text classification: A comprehensive comparative study. *Information Processing Management*, 58(3):102481, 2021.
- [15] Nirdosh Rawal, Prudhvith Tavva, and Prakash Selvakumar. Enhancing large language model performance with reinforcement learning from human feedback: A comprehensive study on qa, summarization, and classification. In *2024 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pages 1–6, 2024.

## A Appendix

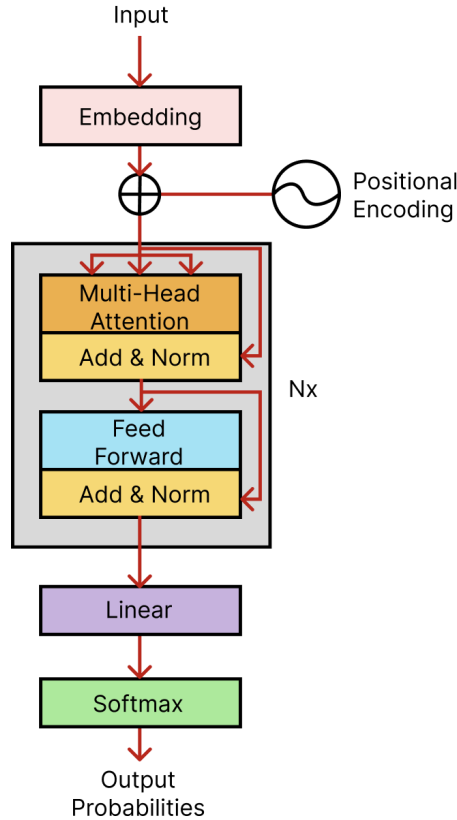


Figure 1: Gemma-2-2B-it architecture

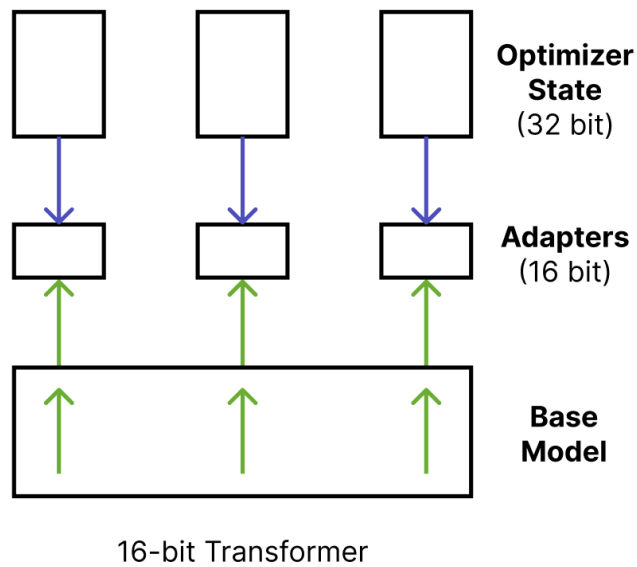





Figure 2: LoRA method and its memory requirements

### Verbosity Bias

 **Prompt:**  
"What are the main causes of climate change?"

 **Response A:**  
"Climate change is primarily caused by the emission of greenhouse gases, notably carbon dioxide, methane, and nitrous oxide, which result from human activities like burning fossil fuels, deforestation, and intensive agriculture."

 **Response B:**  
"The main cause of climate change is human activity."

**Bias Explanation:**  
Despite Response B being concise and accurate, a model suffering from verbosity bias mistakenly **favor Response A** simply because it's longer, regardless of whether users prefer succinctness.

Figure 3: Example: Verbosity Bias

## Position Bias



**Prompt:**  
"What is the capital city of Australia?"



**Ordering 1:**  
**Response A:** "Sydney is a major city but not the capital of Australia; Canberra is the capital."  
**Response B:** "Canberra is the capital of Australia."



**Ordering 2:**  
**Response A:** "Canberra is the capital of Australia."  
**Response B:** "Sydney is a major city but not the capital of Australia; Canberra is the capital."

### Bias Explanation:

Position bias occurs if the model consistently favors the earlier response regardless of content. Ideally, the prediction of human preference should remain stable when responses switch order. A biased model might **disproportionately prefer the first response** in each scenario.

Figure 4: Example: Position Bias

## Self-Enhancement Bias



**Prompt:**  
"Can antibiotics cure viral infections?"



**Response A (Overly confident but incorrect):**  
"Yes, antibiotics are highly effective at curing all kinds of infections, including viruses."



**Response B (Accurate but cautious):**  
"No, antibiotics are effective against bacterial infections but do not cure viral infections."

### Bias Explanation:

A model influenced by self-enhancement bias might mistakenly **favor the overly confident Response A**, despite its factual inaccuracy, due to its assertive tone, rather than the cautious but correct Response B.

Figure 5: Example: Self-enhancement Bias