Enhancing LLM Performance via LoRA SFT and DPO for the LM Playschool Challenge

Vasamsetti Nihal Tej
IIT Bhubaneswar
23cs01071@iitbbs.ac.in

Velineni Harshavaishnav
IIT Bhubaneswar
23cs01067@iitbbs.ac.in

Shreya Ghosh
IIT Bhubaneswar
shreya@iitbbs.ac.in

Abstract

Large Language Models (LLMs) have achieved SOTA results in many natural language processing tasks. However, they are primarily trained for static text generation and often struggle in interactive, goal-oriented environments that require multi-turn reasoning, state tracking, and need to give structured output formats. This paper presents our team's submission to the LM Playschool Workshop Shared Task at EMNLP 2026. The LM Playschool Workshop Shared Task at EMNLP 2026 addresses this challenge by evaluating language models as interactive agents in structured dialogue games. We hereby explored different fine-tuning strategies for Qwen 4B and Qwen 7B models to improve their interactive capabilities and evaluated them. We perform a comparison of Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO) using multi-turn training data. Our study analyzes the effectiveness of each optimization strategy for learning robust dialogue behaviors while maintaining models stability towards interactive games and also not to forget its language linguistics by different evaluating metrics. Experimental results show that although supervised fine-tuning provides consistent interactive performance, directly optimizing the foundation model using DPO reduces optimization errors and leads to improved agent behavior. Our best-performing model achieved a **clemscore of 38.23** on the official LM Playschool leaderboard, evaluated using the Playpen workbench.

1 Introduction

Most Large Language Models (LLMs) are pretrained on static text corpora and are not designed to learn or operate effectively in interactive environments. Although instruction tuning aligns LLMs for single-turn, open-ended tasks, they still struggle to function as goal-oriented interactive agents. As, Multi-turn dialogue games require models to maintain dialogue history, accurately track game states, follow structural output formats, and also need to satisfy environment-specific constraints throughout long interactions to maintain correctness.

One major challenge is that pretrained LLMs often fail to consistently follow the required response format specified by the user or the environment, we need to also ensure model may lose its language knowledge when training for interactive games. Dialogue games provide an effective benchmark for evaluating and improving these interactive capabilities, these workbenches expect a strict output format in the interactions else they may go into abort/lose state.

The LM Playschool Shared Task, built on the Playpen-2025 benchmark, evaluates agentic capabilities through conversational board games and dialogue games using the `clembench` workbench. Models are required to correctly interpret user feedback, maintain implicit game states across multiple turns, and generate concise, structured responses that strictly follow the required output format, such as `Instruction: [Action]` followed by state markers like `DONE`.

In this work, we present our complete training pipeline and the techniques used to improve interactive dialogue capabilities. During development, we encountered two major challenges: handling multi-turn dialogue data and designing effective data processing pipelines, and optimizing fine-tuning strategies. Our main contributions are summarized as follows:

1. We design a data processing pipeline to handle multi-turn dialogue interactions and generate training data for SFT and DPO. Since each training method requires a different data format, we develop separate preprocessing and data generation pipelines for each approach.

2. We further analyze the effect of different hyperparameter settings on the interactive performance of these models by evaluating with

the playpen workbench, which contains two primary scores: clemscore and statscore.

2 Data Preprocessing and Engineering Pipeline

Our primary training corpus is derived from the raw interaction traces of the Playpen-2025 benchmark. The benchmark contains interaction logs collected across four different versions of the Playpen environment. Each game session is stored as a complete multi-turn conversation under the `chat` field, represented as a list of nested dictionaries containing user and assistant messages. The complete dataset consists of **114,479 interaction traces**.

Since each optimization technique requires a different data format, we design separate preprocessing pipelines for SFT and DPO. Additionally, we perform a warm-up fine-tuning stage to familiarize the model with interactive dialogue generation.During this phase, we train the model using warmup data made up of strict, rule-based questions and answers. This teaches the model the basic rules of chatting, making sure it knows how to follow exact formats and instructions before it moves on to harder, more open-ended conversations. During this stage, the model is trained using a small learning rate to learn the required response format and follow the output structure specified by the input before proceeding to the main fine-tuning process.

2.1 Supervised Fine-Tuning (SFT)

We first filter the dataset to retain only successful interactions, resulting in **31,779 successful game sessions**. Rather than training only on the final successful responses, we generate intermediate training examples using a cumulative dialogue prefix extraction algorithm.

Let a dialogue session be represented as

$$S = \{u_1, a_1, u_2, a_2, \ldots, u_T, a_T\}, \tag{1}$$

where u_t denotes the user (or environment) message and a_t denotes the assistant response.

For every assistant turn, we generate one training example by retaining the dialogue history up to and including the corresponding assistant response:

$$x_t = [u_1, a_1, \ldots, u_t, a_t]. \tag{2}$$

During supervised fine-tuning, only the assistant tokens in a_t contribute to the causal language modeling loss. This cumulative dialogue prefix strategy converts each successful dialogue into multiple training examples while preserving the complete conversational context. After using this strategy, we got **60,011 interactions**.The warm-up data is taken up from the playpen-paper-2025 which contains **20,266 interaction traces**.Both of these datasets are combined and preprocessed using Qwen3.5 tokenizer. Upon removing the duplicates, our dataset has been reduced to **58,088 interactions**.

2.2 Direct Preference Optimization (DPO)

We first filter the dataset to retain only successful interactions, resulting in **31,779 successful game sessions.** To prepare the data for Direct Preference Optimization (DPO), the interaction logs are then grouped by the same `game` , `experiment` and `episode` identifiers from the **114,479 interactions**, ensuring that each preference pair is constructed from identical dialogue contexts. Successful responses are treated as preferred responses (y^+), whereas responses from *Lose* or *Aborted* episodes are treated as rejected responses (y^-). Each preference example is represented as

$$(x,\ y^+,\ y^-), \tag{3}$$

where x denotes the shared prompt, y^+ is the preferred response, and y^- is the rejected response. These prompt-preference triplets are then used to optimize the model using the DPO objective. Here,we set the positive-to-negative ratio for the DPO preference pairs to **1:2** which makes it to a total of **63,558** interactions.This dataset is preprocessed with Qwen3.5 tokenizer. Upon removing the duplicates, our dataset has been reduced to **59,154 interactions**.DPO mathematically embeds the reward function directly within the model's token log-probabilities, eliminating the need for an external reward network. Utilizing this metadata bypasses costly online sampling and multi-model hosting, ensuring highly efficient and stable optimization during multi-response comparison.

3 Methodology

Our model adaptation strategy is strictly divided into two phases, ensuring that the model first learns the desired output format before optimizing for preference alignment.

3.1 Low-Rank Adaptation (LoRA)

To reduce the memory overhead of updating billions of parameters, we used LoRA. LoRA freezes the pre-trained model weights $W_0 \in \mathbb{R}^{d\times k}$ and adds trainable rank decomposition matrices to every layer of Transformer architecture. The forward pass is modified as follows for a certain layer:

$$h = W_0 x + \Delta W x = W_0 x + BAx \tag{4}$$

where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$, and the rank $r \ll \min(d, k)$. We applied LoRA adapters mostly to the query and value projection matrices within the self-attention modules.

3.2 Supervised Fine-Tuning (SFT)

In the first phase, we trained the base model using standard autoregressive language modeling. Given a dataset of instruction-response pairs $\mathcal{D}_{\text{SFT}} = \{(x_i, y_i)\}_{i=1}^{N}$, we optimize the negative log-likelihood:

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\sum_{t=1}^{|y|} \log P_\theta(y_t \mid x, y_{<t}) \right] \tag{5}$$

Only the LoRA parameters θ were updated during this phase.

3.3 Direct Preference Optimization (DPO)

As an alternative adaptation strategy, we fine-tuned the base model using DPO. Unlike Reinforcement Learning from Human Feedback (RLHF), which relies on training a separate reward model followed by reinforcement learning, DPO directly learns from pairwise preference data. For each prompt x, the training dataset contains a preferred response y_w and a rejected response y_l. This technique encourages the model to assign a higher probability to the preferred response while reducing the likelihood of the rejected response, thereby improving alignment with human preferences. Additionally, DPO regularizes the optimized policy to remain close to the reference model, resulting in an optimized approach that is both computationally efficient and stable. The DPO loss is defined as:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x,y_w,y_l)} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right] \tag{6}$$

Here, π_{ref} denotes the frozen reference model, π_θ is the DPO model being trained, and β controls the deviation from the reference policy. The DPO model is fine-tuned using preference pairs consisting of chosen and rejected responses.

4 Experimental Setup

4.1 Base Model Architecture

- **Base Model:** Qwen3.5-4B
- **Parameters:** 4 Billion

4.2 Optimization Details

The optimization hyperparameters for both phases were carefully tuned to prevent catastrophic forgetting. We also used statscore to diagnose the catastrophic forgetting.

- **Context Length:** We set the maximum context length to 2048 for SFT and 1024 for DPO.
- **LoRA Configuration:** Rank (r) = 32 for SFT and 16 for DPO, Alpha (α) = 64 for SFT, 32 for DPO, Dropout = 0.05.
- **Batch Size:** [e.g., Global batch size of 16] (per device batch size of 2 with 8 gradient accumulation steps).
- **Learning Rate:** 2e-4 for SFT, 5e-5 for DPO (Cosine scheduler with linear warmup).
- **Epochs:** 1 for SFT, 1 for DPO.
- **Optimizer:** AdamW ($\beta = 0.1$, weight decay=0.01).
- **Compute Budget:** 1x NVIDIA RTX 5000 Ada Generation 32GB GPU for approximately 13 hours total.
- **Random Seed:** 42.

Mixed-precision training (bfloat16) was utilized to accelerate training and reduce VRAM usage.

4.3 Inference and Decoding Strategy

During final evaluation on the LM Playschool framework, we employed the following deterministic decoding parameters to ensure reproducible generation:

- **Temperature:** 0.0
- **Max New Tokens:** 300

Model	Clem	Δ	Stat	Δ
Qwen3.5-4B				
Base Pre-trained	37.33	0.00	52.07	0.00
LoRA SFT	38.03	+0.70	48.07	-4.00
LoRA DPO (Final)	**38.23**	+0.90	53.56	+1.49
Qwen2.5-7B-Instruct				
Base Pre-trained	26.52	0.00	48.35	0.00
LoRA SFT	34.23	+7.52	49.67	+1.32

Table 1: Validation performance across training stages for Qwen3.5 and Qwen2.5 architectures.

5 Results and Evaluation

We evaluated our models strictly according to the LM Playschool Challenge guidelines, relying on the *clemscore* and *statscore* as our primary metrics for conversational capability and task adherence. Briefly, the **clemscore** measures the model's interactive conversational capabilities and its ability to follow fine-grained rules within simulated, multi-turn dialogue games (derived from the clembench framework). In contrast, the **statscore** acts as a baseline for general knowledge and reasoning by quantifying the model's aggregate performance across a suite of standard and static benchmarks. All results are evaluated only in a single run.

As shown in Table 1, the SFT phase successfully aligned the base model to follow instructions, achieving a solid baseline clemscore of 38.03. The DPO phase successfully refined the model's outputs by penalizing undesirable generation patterns, yielding an improved final score of 38.23.

6 Reproducibility Information

To ensure full transparency and allow reviewers to reproduce our submission, we have open-sourced all necessary models. The models are hosted on the Hugging Face Hub.

- **Final DPO Checkpoint (Adapters):** https://huggingface.co/harshavaishnav/DPO
- **SFT Checkpoint / Full Submission:** https://huggingface.co/harshavaishnav/LMPlayschool_Submission

The exact final merged models that corresponds to the LMPlayschool_Submission repository are linked above. We have done inference on the final adapters but we are instructed to submit the merged models (base model + LoRA adapters). We have changed the partitioning mechanism from auto to cuda to do inference on a single GPU. Except this, we did not deviate from the public evaluation setup provided by the organizers.

7 Conclusion and Future Work

This technical report details our system development for the LM Playschool Workshop Shared Task at EMNLP 2026. Through rigorous data parsing and a systematic exploration of post-training paradigms across Qwen models, we demonstrated that direct preference tuning provides a powerful mechanism for stabilizing agentic behavior in multi-turn games. The stability of the statscore indicates that our approach effectively prevents catastrophic forgetting of the model. Our best run yielded a top leaderboard score of 38.23. Future work will investigate online RL loops (such as PPO or ReST) to integrate real-time game state engine rewards directly into the token optimization process.

Limitations

Despite the improvements, the model still exhibits limitations common to models of this scale, including occasional hallucinations in multi-turn reasoning tasks. The chosen rank ($r = 16$) for LoRA may also bottleneck the model's capacity to internalize entirely new factual knowledge, restricting it primarily to style and format alignment.

References

[1] K. Chalamalasetti, J. Götze, S. Hakimov, et al. 2023. clembench: Using Game Play to Evaluate Chat-Optimized Language Models as Conversational Agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

[2] Horst, Nicola, et al. 2025. Playpen: An Environment for Exploring Learning Through Conversational Interaction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

[3] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*.

[4] L. Ouyang, J. Wu, X. Jiang, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

[5] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *Advances in Neural Information Processing Systems*, 36.

[6] A. Yang, A. Li, B. Yang, et al. 2025. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*.