Proceedings of the 4th International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/2026.AS32439

Building a Domain-Specialized Medical LLM: Pretraining and Multi-Stage Alignment (SFT, PPO, DPO, KTO) for Chinese Medicine

Junqi Yang

Department of Computer Sciences, University of Wisconsin–Madison, Madison, United States, jyang732@wisc,.edu

Abstract: This project studies how far a small medical language model can be improved with a staged alignment pipeline under limited compute. Starting from Qwen3-0.6B, I first continued pretraining on a Traditional Chinese Medicine (TCM) corpus to add domain vocabulary and clinical-style text patterns. I then trained an SFT model on Chinese TCM instruction-response data and built two follow-up branches: an RLHF-style branch with a reward model and PPO, and a preference-only branch with DPO followed by KTO.

All models were evaluated on shared category-level scores from MedBench. In my experiments, the SFT model remained competitive on factual and reasoning-heavy tasks, while the DPO+KTO branch produced more conservative responses and showed better behavior on open-ended prompts. The PPO branch was harder to stabilize on a small backbone and did not consistently outperform SFT on the categories reported here. These results suggest that for a 0.6B medical model, preference alignment is useful, but its effect depends strongly on how the preference data are constructed and how much capacity the base model has.

Keywords: Medical Large Language Models; Model Alignment; Reinforcement Learning from Human Feedback; Direct Preference Optimization; Kahneman–Tversky Optimization; Traditional Chinese Medicine

1. Introduction

Large language models are now used in many domain-specific tasks, including medicine. However, a medical model is not useful only because it can generate fluent answers. It also needs to stay close to reliable medical knowledge, avoid overconfident claims, and respond in a safe and controlled way. These issues become harder when the base model is small and the training budget is limited.

A common way to improve model behavior is to start with supervised fine-tuning and then add an alignment stage. RLHF is one widely studied option, but it usually requires reward modeling and policy optimization, which makes the pipeline more complex and harder to tune in practice [1, 2]. Prior work has also pointed out that reinforcement-based alignment can become unstable and sensitive to reward quality [3]. More recent preference-based methods such as DPO simplify this process by avoiding explicit reinforcement learning [4]. Related work such as SPA and KTO further suggests that useful alignment signals can be learned without a full RL pipeline [5].

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.AS32439

In the medical domain, supervised tuning has already been shown to improve factuality and domain adaptation [6]. Still, it is not obvious whether a compact model also benefits from additional alignment after SFT, or whether those later stages mainly change response style without improving correctness. This question is especially relevant for compact Chinese medical models, where compute is limited and preference data are often synthetic or weakly supervised.

In this work, I study this question on Qwen3-0.6B. I first adapt the model to Chinese medical question answering with SFT. I then compare two additional alignment routes: an SFT–RM–PPO pipeline and an SFT–DPO–KTO pipeline. All three systems are evaluated on MedBench using the category-level metrics shared across models. Rather than asking which method is universally best, I focus on the practical trade-offs they create for a small medical model, including factuality, reasoning, response style, and safety-related behavior.

2. Pretraining

2.1. Model architecture

Our work builds upon Qwen3-0.6B, a dense decoder-only transformer belonging to the latest generation of the Qwen series. The model contains approximately 0.6 billion parameters, including 28 transformer layers, grouped-query attention with 16 query heads and 8 key–value heads, and a maximum context window of 32,768 tokens. As part of the Qwen3 family, the model supports both "thinking" and "non-thinking" modes and possesses strong multilingual capability and robust general-domain reasoning. We use the original model architecture without any modification. We initialize the model with the officially released pretrained checkpoint. This checkpoint is used for the following domain-specific pretraining stage.

2.2. Pretraining data

To improve the model's understanding of Traditional Chinese Medicine (TCM), we use a domain-specific pretraining corpus from the publicly available `tcm_pretrain_corpus` on ModelScope. This dataset contains classical TCM books, materia medica descriptions, symptom–syndrome explanations, clinical case records, modern textbook chapters, and other medical documents. Different from general web-scale corpora used in foundation model pretraining, this dataset contains more concentrated and accurate domain knowledge. It covers meridian theory, qi–blood relations, pattern differentiation, herbal combinations, and TCM diagnostic principles. Most of this knowledge does not appear often in general pretraining data. Therefore, domain adaptation is necessary for stable and consistent TCM reasoning.

Before training, we tokenize all documents with the official Qwen3 tokenizer. We then split long texts into sequences with a maximum length of 2,048 tokens. We do not add any instruction format or conversation format. The model only sees plain text sequences under a standard language modeling objective. We also fully shuffle the dataset. This helps remove ordering bias and increases data diversity.

2.3. Pretraining procedure

We continue pretraining from the Qwen3-0.6B checkpoint. The training objective is standard next-token prediction. All model parameters are updated during this stage. We train the model for one epoch. The learning rate is set to 2×10^{-4}. Cosine decay is used together with a short warmup. We

DOI: 10.54254/2755-2721/2026.AS32439

also apply bf16 precision and gradient accumulation to make training more efficient under limited GPU memory.

After this stage, we obtain a domain-adapted checkpoint. The model still keeps its general reasoning ability. At the same time, it shows better understanding of TCM terms and clinical logic. This checkpoint is later used for supervised fine-tuning and alignment.

3. Supervised Fine-Tuning

After pretraining, we further adapt the model to clinical tasks using supervised fine-tuning (SFT). The pretrained Qwen3-0.6B model already has good language ability and general biomedical knowledge. However, it does not naturally follow instructions. It also does not produce outputs in a typical TCM clinical style. So we use SFT to guide the model toward TCM-style reasoning. This includes tasks such as syndrome differentiation, explanation of prescriptions, and patient-oriented responses.

3.1. Dataset construction

We use the HWT-CM-SFT-v1 dataset for fine-tuning. This is an instruction–response dataset designed for TCM scenarios. Each sample contains an *instruction*, sometimes an *input*, and an *output*. The output is a TCM-based explanation or answer. The dataset covers different areas, such as internal medicine, gynecology, acupuncture, classical prescriptions, constitution analysis, and symptom-pattern matching. Because the dataset is built for TCM, it helps the model learn domain-specific terms and reasoning patterns that are not well learned from general pretraining data.

To control overfitting, we only use the first 5,000 high-quality samples. All samples are processed with the official Qwen3 tokenizer. This keeps the token format consistent across stages.

3.2. Fine-Tuning procedure

We run supervised fine-tuning using the LLaMA-Factory framework. We adopt a LoRA-based method to reduce training cost. In this setup, only a small number of low-rank parameters are updated. Most of the original Qwen3-0.6B weights remain fixed. This helps keep the knowledge learned in pretraining.

Training is done on a single NVIDIA GPU. We follow the hyperparameters defined in the SFT configuration file. The maximum sequence length is set to 1024 tokens. This allows the model to handle relatively long TCM reasoning processes. We use a learning rate of 1×10^{-4}, a batch size of 1, gradient accumulation of 8, and cosine learning-rate scheduling with warmup. The model is trained for 3 epochs. This setting balances convergence speed and the risk of overfitting.

After training, we merge the LoRA weights into the base model. This gives a complete SFT checkpoint. This SFT model is used as the starting point for the later alignment stages, including Reward Modeling, PPO, DPO, and KTO. Through these steps, the general-purpose Qwen3-0.6B model becomes a TCM-oriented instruction-following system.

4. Reward modeling

After SFT, we train a reward model to score candidate responses. The goal of this model is simple. Given two answers to the same medical question, it should give a higher score to the better one. We mainly use it as an intermediate component for later alignment, especially PPO. Different from SFT, this stage does not train the model to imitate reference answers token by token. Instead, it learns a rough preference signal based on response quality under our data construction setting.

DOI: 10.54254/2755-2721/2026.AS32439

4.1. Dataset construction

To build the preference dataset for reward modeling, we use a two-stage generation-and-pairing process. For each TCM diagnostic question, a weak baseline model (Qwen2.5-0.5B) generates a negative response. The positive response is taken from the high-quality HWT-CM-SFT-v1 dataset used in supervised fine-tuning. In this way, we obtain (chosen, rejected) pairs. These pairs match expert-grounded TCM answers with intentionally low-quality outputs. This helps the reward model learn a stable preference for domain-appropriate reasoning. All samples are tokenized with the official Qwen3 tokenizer for compatibility.

4.2. Reward model training procedure

We train the reward model with the LLaMA-Factory pipeline. The hyperparameters are defined in the reward configuration file. We use a lightweight and parameter-efficient LoRA strategy. This allows the model to learn evaluation behavior without overwriting the biomedical and language knowledge from pretraining and SFT.

The reward model is initialized from the Qwen3-0.6B-SFT checkpoint. This makes its evaluation more consistent with the same linguistic and clinical distribution as the supervised model. Only a small number of low-rank adapter weights are updated during training. This reduces computational cost and keeps the backbone model stable.

Training is performed on a single NVIDIA GPU with the settings in the YAML configuration. The learning rate is 1×10^{-5}. The batch size is 1. We use cosine learning-rate scheduling with warmup, gradient accumulation of 16, and a maximum sequence length of 1024 tokens. The model is trained for 2 epochs. This setting balances reward-model stability and the risk of overfitting to the constructed preference pairs.

After training, we merge the LoRA weights into the base model to obtain the final reward model checkpoint. This reward model is used in later alignment stages, especially PPO. In PPO, the policy model updates its responses to maximize the learned reward signal.

5. Reinforcement learning with PPO

After SFT and reward model training, we run one PPO stage to test whether reward-driven optimization can further improve the model. The policy model is initialized from the SFT checkpoint. The reward model is used to score generated responses during training. In this setting, PPO is used to go beyond simple imitation learning. We want to see whether this extra optimization can help a small medical model generate better answers, or whether it mainly changes style and confidence.

5.1. Dataset construction

The PPO dataset comes from a publicly available Chinese medical dialogue corpus on ModelScope [7]. It contains many instruction–response pairs from general medical and TCM scenarios. To keep the experiment feasible, we randomly select 800 samples, as defined in the PPO configuration file. Each sample contains a natural user instruction or question. Some samples also include additional context. Different from the SFT stage, PPO only needs the input side. The responses are generated online by the policy model during rollouts.

All samples are processed with the Qwen3 tokenizer under the Qwen3_nothink template. This keeps the format consistent with previous training stages. The maximum sequence length is 1024

DOI: 10.54254/2755-2721/2026.AS32439

tokens. This setting balances long-form TCM reasoning ability and memory limits. We use 16 workers for parallel preprocessing to speed up the data pipeline.

5.2. PPO training procedure

PPO training starts from the SFT checkpoint instead of the original base model. For each input, the model first generates a response. Then the reward model gives a score. After that, the PPO objective updates the policy with bounded steps. We use the LLaMA-Factory PPO pipeline with LoRA, so only the adapter weights are updated. This setting makes the experiment feasible under limited hardware conditions.

The main hyperparameters follow the PPO configuration file: learning rate 1×10^{-5}, batch size 1, gradient accumulation 8, cosine scheduling with warmup, and one training epoch. The generation length is capped at 128 tokens during rollouts, with nucleus sampling ($p = 0.9$). In practice, this stage is more sensitive than SFT. Because the base model is small and the reward signal is only an approximation of answer quality, PPO can improve fluency or apparent helpfulness without reliably improving factual correctness.

6. Direct Preference Optimization (DPO)

Following supervised fine-tuning, we apply Direct Preference Optimization (DPO) to further align the model with human preferences. Unlike PPO-based alignment, which relies on reinforcement learning and an explicit reward model, DPO directly optimizes a pairwise preference objective using chosen–rejected response pairs. This yields a stable and compute-efficient alignment stage and forms a second branch in our pipeline, resulting in the SFT $\rightarrow$ DPO $\rightarrow$ KTO model alongside the SFT–RM–PPO branch.

DPO is well suited to the TCM domain, as it avoids free-form rollouts and naturally reuses the preference pairs constructed during reward-model data preparation.

6.1. Dataset construction

The DPO dataset used in this work is derived directly from the preference data prepared during the reward-model training stage. Each instruction sample contains a *chosen* response representing a clinically appropriate TCM answer, and a *rejected* response typically generated by a weaker model (Qwen2.5-0.5B-Instruct). The rejected answers often contain insufficient clinical reasoning, missing diagnostic cues, or incomplete treatment logic, which makes them effective negative samples for preference learning.

Only the first 1,000 instruction–response pairs were selected for DPO training in order to maintain a balance between dataset diversity and computational cost. Each sample consists of an instruction followed by its paired chosen and rejected responses. All text was tokenized using the Qwen3 tokenizer with a maximum input length of 2048 tokens, consistent with the configuration used in earlier stages.

6.2. Training procedure

DPO was trained using the LLaMA-Factory framework with LoRA fine-tuning. The pretrained SFT checkpoint (`Qwen3_pt_sft`) was used as the initialization, ensuring that preference alignment is applied on top of a domain-specialized TCM model.

Only LoRA parameters were updated during training while the base Qwen3-0.6B backbone remained frozen. The training configuration included a learning rate of `5e-6`, batch size 1 with gradient

DOI: 10.54254/2755-2721/2026.AS32439

accumulation of 8, cosine learning-rate scheduling, and a total of three epochs. This configuration keeps memory usage low and avoids instability typically associated with reinforcement-learning-based methods.

Compared with PPO, which requires sampling model rollouts and computing reward-based advantages, DPO provides a deterministic and stable optimization process. This is particularly advantageous in the medical TCM setting, where hallucinated rollouts may introduce unsafe content and where monotonic preference improvement is desired. The resulting DPO-aligned model serves as the initialization for the subsequent KTO training stage.

7. Kahneman–Tversky Optimization (KTO)

Following supervised fine-tuning and preference alignment, we apply Kahneman–Tversky Optimization (KTO) as the final alignment stage. Inspired by prospect theory, KTO introduces an asymmetric objective in which dispreferred outputs are penalized more strongly than preferred outputs are rewarded, yielding a risk-averse training signal around well-aligned behavior.

Unlike PPO-based RLHF, KTO does not need an explicit reward model or policy rollout. Each training sample has a binary `kto_tag` to show whether the response is acceptable. The optimization is done directly on response log-probabilities under a prospect-theoretic utility. In our pipeline, KTO is applied after the SFT→DPO branch. Its goal is to further refine preference-aligned behavior and reduce clinically unsafe or logically inconsistent outputs.

7.1. Dataset construction

For KTO training, we build the dataset from the ShenNong-TCM-LLM corpus. This corpus contains expert-written instruction–response pairs from major TCM subdomains. Each sample follows the LLaMA-Factory format. It also includes a binary `kto_tag` to indicate whether the model-generated response meets basic correctness and safety requirements. To balance computational cost and domain coverage, we uniformly sample about 2,000 instances from different TCM categories. All texts are tokenized with the Qwen3 tokenizer. The maximum sequence length is set to 2,048 tokens.

7.2. Training procedure

KTO training starts from the DPO checkpoint. We continue training with LoRA-based fine-tuning in LLaMA-Factory. The base parameters of Qwen3-0.6B stay frozen, and only the adapter weights are updated. We use the hyperparameters defined in the KTO configuration file. The learning rate is 5×10^{-6}. The batch size is 1. The gradient accumulation step is 8. We use cosine scheduling with warmup. The model is trained for three epochs.

The main purpose of this stage is not to introduce another reward model. Instead, it adjusts the model with a binary preference signal. In our setting, this objective makes the model more cautious. It reduces some low-quality or unsafe responses. However, it can also make the answers shorter. In some cases, the model becomes less willing to give a clear medical judgment when the prompt needs a specific answer.

8. Evaluation

We evaluate the SFT, SFT–RM–PPO, and SFT–DPO–KTO models on the MedBench benchmark. This benchmark measures several aspects of clinical ability, including knowledge recall, language generation, reasoning, understanding, and safety. Because some MedBench sub-tasks are not shared by all three models, we only report category-level metrics that are available for every model. This

DOI: 10.54254/2755-2721/2026.AS32439

makes the comparison more consistent and fair. It also helps us focus on the effect of each training method.

8.1. Medical knowledge question answering

Table1. Medical Knowledge Question Answering Scores

Model	Score
SFT	32.9
PPO	29.6
KTO	29.6

From Table 1, SFT has the highest score, while PPO and KTO show the same lower result. This is expected. The task mainly prefers short and correct answers. SFT is trained directly on this type of data.

The PPO model often produces longer responses. These answers may look more detailed. However, the extra content does not always improve correctness. In some cases, the model sounds more confident, but the medical content is not better.

KTO behaves in another way. Its outputs are usually more cautious. This can reduce clearly wrong answers. At the same time, it may also make some responses incomplete. For this task, incomplete answers are often penalized. So this cautious style does not help much.

Overall, this result is more about response behavior. SFT follows the task format more closely. PPO and KTO change the response style, but not always in a helpful way.

8.2. Medical language generation

Table2. Medical Language Generation Scores

Model	Score
SFT	39.1
PPO	38.5
KTO	38.5

For language generation, PPO gives the most expressive outputs. SFT and KTO are more restricted in comparison. SFT uses supervised training. Its responses are usually clear and correct. However, they often follow fixed patterns from the dataset. Because of this, the outputs may look less flexible.

PPO changes the behavior of the model. It encourages longer and more confident responses. This comes from the reward signal, which often prefers answers that look complete and helpful. As a result, PPO outputs look more polished. But this does not always mean the answers are more accurate. In many cases, only the style is improved.

DOI: 10.54254/2755-2721/2026.AS32439

KTO takes a more conservative approach. Its loss function penalizes unsafe or low-quality outputs. So the model tends to generate shorter answers. It also avoids making strong claims. This can reduce hallucinations. However, it also makes it harder for the model to give detailed explanations.

In this task, we can see three different behaviors. SFT focuses on correctness. PPO focuses more on expression. KTO focuses on safety.

8.3. Complex medical reasoning

Table3. Complex Medical Reasoning Scores

Model	Score
SFT	23.2
PPO	23.4
KTO	23.4

This part shows a different pattern. The gap between methods is small, but the behavior is not the same. SFT still shows slightly better reasoning ability. One reason is that it is trained on structured question–answer data. These samples include step-by-step explanations. So the model can learn common reasoning patterns and reuse them.

PPO outputs are usually fluent. But fluency does not mean correct reasoning. The reward model mainly evaluates overall quality, not detailed logic. Because of this, PPO may prefer answers that look confident or complete. Sometimes the reasoning chain sounds reasonable but is not medically correct. This problem is often seen in RLHF settings.

KTO shows another limitation. It avoids making strong conclusions unless the evidence is clear. This behavior is safer, but it is not always suitable for reasoning tasks. Medical reasoning often requires making decisions with incomplete information. For a small model like Qwen3-0.6B, this cautious behavior makes the reasoning chain shorter. Some answers stop too early or do not fully explain the logic.

From these results, supervised training still plays an important role in reasoning tasks. Preference-based methods can change behavior, but they do not always improve reasoning quality without careful design.

8.4. Medical language understanding

Table4. Medical Language Understanding Scores

Model	Score
SFT	26.3
PPO	25.0
KTO	25.0

DOI: 10.54254/2755-2721/2026.AS32439

Table 4 again shows that SFT performs best. Tasks in this category require the model to read the input carefully and keep important medical details. From the score pattern, later preference-based alignment does not clearly improve this ability.

A possible reason is that SFT directly trains the model on input–output pairs where specific terms, symptoms, and relations are important. PPO, in contrast, optimizes the model with reward signals after generation. This may affect response style and confidence, but it does not necessarily improve the model's ability to understand the input itself. KTO is more conservative. This may reduce hallucinated details, but it can also make the model leave out useful information that should have been extracted.

So the main takeaway from this section is simple. For medical tasks that depend heavily on language understanding, the supervised stage seems to contribute most of the useful improvement in this project.

8.5. Medical safety and ethics

Table5. Medical Safety and Ethics Scores

Model	Score
SFT	3.3
PPO	3.3
KTO	3.3

Table 5 shows no score gap across the three models in this category. All of them receive 3.3, so this section should be interpreted carefully. Based on the benchmark numbers alone, there is no clear winner on safety and ethics.

That said, the models still differ in response style. In my inspection, KTO tends to be more cautious and is less willing to make strong claims when the prompt is ambiguous. SFT is often more direct, which can be helpful when the answer is clear but may sound too certain in borderline cases. PPO responses are usually more elaborate, and that extra fluency can sometimes make weak reasoning sound more convincing than it is.

Because the benchmark score is tied, I do not treat this category as evidence that one alignment method is definitively safer than the others. A stronger conclusion would require example-level analysis or a more targeted safety evaluation.

9. Conclusion

In this project, we study three training strategies for a small Chinese medical language model. These include SFT only, SFT combined with reward modeling and PPO, and SFT combined with DPO and KTO. Based on the shared MedBench categories, SFT still gives a stable baseline. It works well when the task focuses on direct factual answers and follows the format seen in supervised data.

The other two training routes change the model behavior in different ways. PPO tends to make the responses longer and smoother. In some cases, the answers look more complete. However, this change does not always lead to better benchmark scores. KTO shows a different pattern. It often makes the model more cautious. This can be helpful in open-ended questions, but it may also cause the model to give shorter or less complete answers. For a small model like 0.6B, these effects can already be observed even if the score differences are not large.

DOI: 10.54254/2755-2721/2026.AS32439

From these results, a simple takeaway is that additional alignment after SFT does not always lead to clear improvements. In some cases, it only changes how the model responds. The final effect depends on several factors, such as the quality of the preference signal, the type of benchmark, and the limited capacity of the base model.

10. Limitations

This work has several limitations. First, the base model is relatively small compared with recent medical LLMs. This limits its ability in complex reasoning and clinical abstraction. It may also create a performance bottleneck that is not directly caused by the alignment method itself.

Second, the pretraining and supervised fine-tuning datasets are domain-specific, but they are not fully curated under a uniform clinical safety standard. Because of this, the data may contain biases, outdated medical knowledge, or style-related artifacts. More carefully validated datasets would likely improve the model's robustness and factual reliability.

Third, both PPO and KTO depend on surrogate preference signals. These signals come from reward models or synthetic preference pairs. They may not fully match real expert medical judgment. If clinician feedback can be added in the loop, the alignment quality would likely be more reliable.

Finally, the evaluation is limited to MedBench and mainly focuses on Chinese medical question answering. More benchmarks are still needed to test the model more comprehensively. For example, future evaluation can include multimodal reasoning, long-context behavior, conversational safety, and adversarial robustness. These aspects are important for judging whether the model is ready for real clinical use.

References

[1] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

[2] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. In *International Conference on Learning Representations (ICLR)*, 2017.

[3] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. *arXiv preprint arXiv:2309.11432*, 2023.

[4] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[5] Kawin Ethayarajh, Yan Xu, Dan Jurafsky, and Daniel Khashabi. Supervised preference alignment: A tractable and data-efficient alternative to rlhf. *arXiv preprint arXiv:2412.15115*, 2024.

[6] Karan Singhal, Shekoofeh Azizi, Tao Tu, Saeed Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 2023.

[7] ModelScope. Chinese medical dialogue corpus. https://modelscope.cn/datasets/alexhuangguo/chinese-medical. Accessed: 2026-03-20.