Small-but-Mighty Tech Support LLM

Adaptive Knowledge Distillation + GRPO for Domain-Specialized Language Models (Voice-Enabled Troubleshooting Assistant)

Context:

Large LLMs are strong generalists but expensive to run. This project builds a compact domain-specialized student LLM (≈1–3B params) for consumer-electronics troubleshooting (headphones, speakers, smart devices) that retains language quality. Students will (1) distill knowledge from a larger teacher model, then (2) fine-tune the student with Group Reward Policy Optimization (GRPO) using multi-objective rewards that balance technical accuracy and language quality (grammar, clarity, helpfulness). Finally, they will (3) deliver a voice interface (STT → LLM → UI) and (4) an evaluation suite covering domain expertise and general language retention.

Goal:

- **Distill & Compress:** Implement a KD pipeline from a 7B-class teacher → 1–3B student, focused on device troubleshooting.

- **Align with GRPO:** Design multi-objective rewards (domain correctness + language quality) and train the student with GRPO.

- **Demonstrate:** Ship a voice-enabled demo app where users ask device questions and get step-by-step solutions.

- **Evaluate & Report:** Provide a rigorous evaluation of domain specialization vs. general language retention, with ablations.

Task Suggestions:

A) Literature & Setup

- Read on **KD for LLMs** (response/logit distillation, temperature, domain-aware sampling) and **RL for LLMs** (RLHF, **GRPO**).
- Choose **teacher** (e.g., LLaMA-7B/Qwen-7B-Instruct) and **student** (1.5–3B).
- Curate a **domain corpus**: manuals, troubleshooting guides, vendor FAQs, forum Q&A. Clean, deduplicate, and tag by device/type.

B) KD Implementation

- **Response-based KD:** Generate teacher answers for domain prompts; fine-tune student to imitate (supervised).
- (Optional) **Soft-logit KD:** KL loss with teacher distributions (temperature $\tau>1$).
- **Domain-aware sampling:** Up-sample subdomains where student lags teacher.
- Track **KD metrics:** imitation loss, exact match/ROUGE to teacher, domain QA accuracy (small held-out set).

C) GRPO + Multi-Objective Rewards

- **Reward design:**
  - **R_tech (domain correctness/completeness):**
    - LLM-as-judge (teacher or strong model) scoring 0–10 on correctness/steps.
    - Optional rule bonus for required steps/keywords (e.g., "reset pairing", "firmware update").
  - **R_lang (language quality):**
    - Grammar/fluency score via a grammar API or LLM-as-judge (0–5).
    - Optional structure bonus for numbered steps & clarity.
  - **Total reward:** `R = w_tech * R_tech + w_lang * R_lang` (start 0.6/0.4; tune).
- **GRPO loop:**
    - Sample prompts; generate K responses per prompt with the student.

        - Score each response with R; compute **group-relative advantages**.
        - Policy update (no value net); optional light KL to reference.
- **Tuning knobs:** LR, K per prompt (4–8), length penalty, reward weights, KL strength.
- **Diagnostics:** Reward hacking checks (verbosity, boilerplate, off-topic), diversity, stability.

D) Evaluation & Analysis

- **Domain expertise:**
    - **Technical Accuracy** (task success), **Completeness** (coverage of steps), **Practical Utility** (actionability).
    - Compare Teacher, Student-KD, Student-KD+GRPO.
- **Language retention:**
    - Grammar/fluency score; short general QA/chat sanity checks; perplexity on a generic corpus (small sample).
- **Trade-offs:** Plot accuracy vs. grammar vs. length; analyze failure modes; ablations (no R_lang, no R_tech, different K).
- **Efficiency:** Latency (tokens/s), peak VRAM, model size vs. teacher.

E) Voice Demo (STT → LLM → UI)

- **STT:** Whisper (faster-whisper) with small/medium model; optional lexicon prompts for device names.
- **Backend:** FastAPI + websockets; streaming generation; simple rate limiting.
- **UI:** Minimal web app: mic button, transcript, streaming answer with numbered steps, copy/share.
- (Optional) **TTS:** Piper/Orpheus for read-back; barge-in not required.

References

Core methods & theory

- Hinton, G., Vinyals, O., & Dean, J. (2015). *Distilling the Knowledge in a Neural Network.* arXiv. (arXiv)
- Ouyang, L., et al. (2022). *Training Language Models to Follow Instructions with Human Feedback (InstructGPT).* NeurIPS. (arXiv, NeurIPS Proceedings)
- Guo, D., et al. (2025). *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs (GRPO-based RL).* arXiv. (arXiv)
- Xu, H., et al. (2025). *KDRL: Post-Training Reasoning LLMs via Unified Knowledge Distillation and Reinforcement Learning.* arXiv. (arXiv)
- Li, D., Hao, Y., & Mou, L. (2024). *LLMR: Knowledge Distillation with a Large Language Model-Induced Reward.* LREC/COLING. (ACL Anthology, arXiv)
- Liu, J., et al. (2024). *DDK: Distilling Domain Knowledge for Efficient Large Language Models.* arXiv. (arXiv)
- Stanford CRFM (2023). *Alpaca: A Strong, Replicable Instruction-Following Model (52K instructions).* Blog + repo. (crfm.stanford.edu, GitHub)

RL tooling & libraries

- Hugging Face. *TRL: Transformer Reinforcement Learning (supports GRPO, PPO, DPO, RM, SFT).* Docs / GitHub. (Hugging Face, GitHub)
- CarperAI. *TRLX: Distributed RLHF training for LLMs.* GitHub. (GitHub)
- OxenAI (2025). *Why GRPO is Important and How it Works.* Blog explainer. (Oxen.ai)

Parameter-efficient finetuning & serving

- Hu, E. J., et al. (2021). *LoRA: Low-Rank Adaptation of Large Language Models.* arXiv / OpenReview. (arXiv, OpenReview)
- Hugging Face. *PEFT: Parameter-Efficient Fine-Tuning library.* Blog / Docs / GitHub. (Hugging Face, GitHub)
- Xu, L., et al. (2023). *Parameter-Efficient Fine-Tuning Methods for PLMs (Survey).* arXiv. (arXiv)
- Kwon, W., et al. (2023). *vLLM: Efficient Memory Management for LLM Serving with PagedAttention.* arXiv / ACM. (arXiv, ACM Digital Library)

Speech interface (STT) & utilities

- Radford, A., et al. (2022/2023). *Whisper: Robust Speech Recognition via Large-Scale Weak Supervision.* arXiv / ICML PMLR / GitHub. (arXiv, Proceedings of Machine Learning Research, GitHub)
- SYSTRAN. *faster-whisper (Whisper via CTranslate2, fast inference).* GitHub. (GitHub)
- LanguageTool. *Open-source grammar/style checker (multi-language).* Website / Dev docs. (LanguageTool)
- Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.* arXiv / Docs. (arXiv, sbert.net)

Data & instruction-following resources

- Stanford CRFM (2023). *Alpaca dataset (52K instruction-following samples).* CRFM page / HF dataset. (crfm.stanford.edu, Hugging Face)

Optional background reads (math explainer / overview):

- Samia Şahin (2025). *The Math Behind DeepSeek: A Deep Dive into GRPO.* Medium. (Medium)