Self-Improving LLM Project Plan

Objective

Fine-tune a 7 B student model so it learns to *ask clarifying sub-questions*, harvest GPT-4 "teacher" answers, and distil those chains of thought back into itself (System-2 → System-1). Two task options are offered—pick one based on your evaluation budget and desired focus.

Option 1 StrategyQA (recommended)

Binary yes/no commonsense questions that rely on 2–3 supporting facts.

1. Rationale

- **Short context ➜ low token cost**
- **Legible clarifications** (e.g. "Is the Amazon River in Peru?")
- Single-token gold label simplifies evaluation (Accuracy).

2. Dataset

Split	Size
Train	2 305
Dev	565
Test	490

3. Data-Generation Loop

1. **Sub-sample** \~2 000 train rows.
2. **Prompt template** (≈35 tokens):

```
Q: <original yes/no>
Student draft: <answer + 1–2 clarifying questions>
Teacher (GPT-4): step-by-step THOUGHT + final yes/no
Student rewrite: integrate answer
```

3. Save two parallel corpora:
4. **Track A** – *(Q → answer)* for baseline.
5. **Track B** – *(Q + teacher CoT → answer)* for CoT distillation.

4. Training

- **Baseline**: QLoRA (4-bit), 3 epochs, lr 2e-4.
- **CoT model**: identical hyper-params but using Track B.
- **DPO** *(optional)*: preference pairs (teacher answer > student draft), 1 epoch.

5. Evaluation

```
python strategyqa_evaluator.py --pred <model_preds.json> --gold <dev.jsonl>
```

Target: **+10–15 pp Accuracy** over baseline.

6. Resources & Budget

- GPT-4 calls: \~2 000 × 150 tkn ≈ 300 k tokens → **≈ \$12**.
- GPU: single RTX-4060 (8 GB) fits sequence ≤128 tokens.

7. Timeline (7 days)

Day	Task
1	Repo setup, dataset fetch & sampling
2	Prompt engineering, generation script
3	Run GPT-4 loop, sanity-check outputs
4	Filter & format training sets
5	Train baseline + CoT models
6	Evaluate, inspect errors, run DPO
7	Write report & slides

Option 2 HotpotQA (alternative)

Open-domain, paragraph-level multi-hop QA with supporting-fact supervision.

1. Rationale

Rich retrieval component and explicit supporting facts suit explainability, but **token cost is 3–4× higher** and sub-questions often require external evidence.

2. Dataset (full)

Split	Size
Train	90 447
Dev	7 405
Test	7 405

3. Data-Generation Loop (changes vs. StrategyQA)

1. Sub-sample **4 000** train rows to cap cost.
2. Add retrieval instructions in clarifying questions (“Which page mentions…?”).
3. Retain gold *answer span* and *supporting sentences*—needed for EM/F1 scorer.

4. Training Differences

- Max sequence length 512; gradient-accumulation = 8.
- Memory-efficient attention (Flash-Attn) recommended.

5. Evaluation

```
python hotpot_evaluate_v1.py <gold.json> <pred.json>
```

Report both **Answer EM/F1** and **Supporting-fact EM/F1**.

6. Resources & Budget

- GPT-4 calls: 4 000 × 450 tkn ≈ 1.8 M tokens → **≈ \$70**.
- GPU: 24 GB (e.g. 4090) or gradient checkpointing on 16 GB.
- Retrieval index (FAISS) adds 3 GB RAM.

7. Timeline (10 days)

Day	Task
1–2	Setup, FAISS build, dataset sampling
3–4	Prompt loop with retrieval
5	Clean & align teacher CoTs
6–7	Baseline + CoT training
8	Evaluate, error analysis
9–10	DPO + final report

Quick Comparison

Aspect	StrategyQA	HotpotQA
Token cost	**Low**	High
Context length	≤4 sent.	Paragraphs
Clarification ease	High	Moderate
Eval metric	Accuracy	EM/F1
GPU RAM need	8 GB	16–24 GB
Timeline	7 days	10 days

Deliverables

1. *Data* – processed Track A/B JSONL files.
2. *Models* – baseline, CoT-distilled, (optional) DPO.
3. *Report* – methodology, results, and discussion.
4. *Slide deck* – concise summary for course presentation.

Prepared: 26 July 2025