Problem 3 — Continual Learning (CL)

Jefferson National Lab — Data Scientist Postdoc Interview

1. Literature Review (Decision-Oriented)

1. Replay (Experience Rehearsal): Maintain a small exemplar buffer (or generator). Pros: strong retention, simple, efficient. Cons: stores data or trains generator.
2. Regularization: EWC, Online EWC, and LwF/Knowledge Distillation. Pros: low memory, no raw data needed. Cons: weaker under large shifts, sensitive to hyper-parameters.
3. Architectural Approaches: Adapters, Progressive Nets, task routing. Pros: nearly no forgetting. Cons: parameter growth, more complex serving.

Takeaway: With clear task boundaries and bounded compute/memory, Replay (with KD, optionally EWC) offers the strongest trade-off for accuracy, memory, and compute.

2. Formulated Solution

Chosen approach: **Replay + Knowledge Distillation (KD) + Online EWC**.

Why this combination?

1. Replay: Provides the strongest defense against catastrophic forgetting.
2. Knowledge Distillation (KD): Preserves prior function behavior on current inputs.
3. Online-EWC: Cheap, parameter-space constraint to further stabilize weights.

Training Loop (per task τ)

1. 1. Freeze a teacher model $f\theta^*$ from current $f\theta$.
2. 2. Mix minibatches: 80% current-task, 20% replay buffer.
3. 3. Loss = CE(new labels) + λKD·KD loss + λEWC·EWC penalty.
4. 4. Update weights with AdamW or SGD; early stopping.
5. 5. Update exemplar buffer with class-balanced herding/reservoir.
6. 6. Refresh teacher and Online EWC importance for next task.

Hyper-parameters (robust defaults)

1. Batch mix ratio: 80% current, 20% replay.
2. Exemplar buffer: 100 samples/class (tune 20–200).
3. KD: Temperature=2, λKD=0.5 (tune 0.3–1.0).
4. EWC: λEWC=50 (tune 10–200), decay γ=0.9.
5. Training: 10–30 epochs/task, early stop patience=3.

3. Why this mitigates catastrophic forgetting

1. Replay: Re-exposes past decision boundaries, stabilizing representations.
2. KD: Aligns current logits to prior outputs, preserving function-space behavior.
3. Online-EWC: Constrains parameters critical to previous tasks with negligible cost.

4. Complexity & Resource Balance

1. Training time: Single-model updates with frozen teacher; minutes per task on CPU.
2. Memory: Buffer is the only growing resource (fixed cap M).
3. Compute: No inference overhead; single backbone model serves all tasks.

5. Evaluation Protocol

1. Datasets: Split/Permuted MNIST, Split CIFAR-10/100, domain-specific streams.
2. Metrics: Avg Accuracy, Backward Transfer (BWT), Forward Transfer (FWT), memory/latency footprint.
3. Baselines: Naïve fine-tune, Replay-only, KD-only, Replay+KD, Replay+KD+EWC.

6. Practical Blueprint (first build)

1. Implement data stream iterator and exemplar buffer.
2. Trainer loop with batch mixing, KD, and Online EWC.
3. Evaluation hooks for Avg Acc, BWT, FWT per task.
4. Artifacts: checkpoints, CSV metrics, plots of accuracy matrix and BWT bars.
5. Governance: privacy via feature replay or DP selection; memory budget enforced.

7. Anticipated Panel Q&A;

1. Baseline choice? → Replay buffer; simple, compute-light, strong performance.
2. Privacy concerns? → Use feature replay, KD, or generative replay if raw storage disallowed.
3. Detect forgetting? → Track BWT; values near 0 or positive indicate retention.
4. Task conflicts? → Use KD + EWC; adapters if tasks differ strongly.
5. Production risks? → Monitor drift, checkpoint rollback, strict buffer governance.