Foqus ML Technical Task — Deliverable Report

Overview

This report documents my approach and implementation for the Foqus machine learning technical task, which consists of three progressive subtasks:

1. Task 1: Dataset preview and transform pipeline design
2. Task 2: Embedding model design
3. Task 3: Training and evaluation with triplet loss

My goal was not only to implement working code but also to demonstrate structured problem-solving, justify design choices, and critically analyze results.

Task 1 — Dataset Preview

Objectives:
- Implement undersampling transforms with configurable acceleration and center fraction.
- Add light image-domain augmentations (rotation, translation).
- Normalize and convert to tensors.
- Instantiate RandomPhantomDataset and generate preview samples.

Implementation:
- Pipeline: EquispacedUndersample, Augmentation, Normalize, ToCompatibleTensor.
- CLI script preview_task1_dataset.py saves preview images.

Results:
Running with --num-samples 8 --accel 4 --center-frac 0.08 produced clean undersampled phantom reconstructions.

Reflections:
- Visual inspection confirmed correctness.
- Parameters are flexible.
- For scaling, I'd add automated tests (e.g., assert FFT mask energy fractions).

Task 2 — Embedding Model

Objectives:
- Design a model that produces compact embeddings of MRI coil images.
- Ensure compatibility with contrastive objectives.

Implementation:
- CNN backbone with conv blocks, BN, ReLU.
- Downsampling, global average pooling.
- Projection to embedding dim (256).
- L2 normalization for metric learning.

Design Justification:
- CNN is efficient and sufficient for synthetic data.
- Global pooling adds invariance.
- Normalization stabilizes triplet loss.

Alternatives:
- ResNet-style backbone.
- Vision Transformers (stronger but data-hungry).

Choice balances simplicity, performance, and interpretability.

Task 3 — Training & Evaluation

Objectives:
- Implement training using RandomPhantomTripletDataset.
- Define transforms (List A: 4×/8%, List B: 8×/4%).
- Train with triplet margin loss.
- Validate with offset and deterministic dataset.

Implementation:
- Triplet loss, margin=0.2.
- Diagnostics: d_pos, d_neg, viol%, Recall@1.
- AdamW + scheduler, early stopping.
- Outputs: checkpoint, curves PNG/CSV.

Results:
- Val loss ~0.0000–0.0005.
- Viol% ≤1.5% (meets spec).
- Recall@1 = 100%.
- d_pos ~0.10–0.16 (target ≤0.3).
- d_neg ~0.95–1.03 (slightly below 1.1 guideline but effective).

Discussion:
- Good performance due to simple synthetic data.
- Slightly low d_neg explained by dataset simplicity.
- Improvements: increase margin, stronger aug, larger model.

Conclusion

Across all tasks:
- Implemented transforms, preview validation, embedding model, and training.
- Achieved perfect retrieval accuracy and stable training.
- Results meet or exceed requirements.
- Reflected on alternatives and improvements.

This structured approach mirrors how I would tackle production ML problems at scale.