# Phase 4: Mixed-Precision PTQ on CPU (Weeks 5–6)

## Objective

The objective of Phase 4 was to apply mixed-precision post-training quantization (PTQ) on the preprocessed MobileCLIP model to achieve a Pareto improvement—reducing model size and inference latency while maintaining near-baseline retrieval accuracy and alignment drift. This phase builds directly on the SplitQuantV2 preprocessing completed in Phase 3.

## Methods

Three quantization schedules were designed based on the layer sensitivity map obtained in Phase 2 and the preprocessing results from Phase 3:

• Conservative: fragile = INT8, medium = INT8, robust = INT8 (baseline)  
• Balanced: fragile = INT8, medium = INT8, robust = INT8 (selective quantization)  
• Aggressive: fragile = INT8, medium = INT8, robust = INT8 (quantize all possible layers)  
  
In practice, ONNX Runtime dynamic quantization was used, which supports INT8 quantization on Linear and Gemm operations. While this does not provide sub-8-bit precision, it allowed implementation of schedule-based selective quantization policies for comparison on CPU.

## Evaluation Setup

The quantized models were evaluated on a CPU-only environment with single-thread ONNX Runtime inference. The evaluation dataset consisted of 200 (image, caption) pairs from Flickr30K. Metrics included model size (MB), latency per image–text pair (ms), throughput (samples/s), retrieval accuracy (Recall@1, Recall@5), and mean cosine drift between FP32 and INT8 embeddings.

## Results

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Schedule | Size (MB) | Latency (ms/pair) | Recall@1 | Recall@5 | Mean Cosine Drift | Observation |
| Conservative | 262.5 | 453.7 | 0.0 | 1.6 | 0.186 | Baseline quantization setup. |
| Balanced | 262.5 | 448.7 | 0.0 | 1.6 | 0.186 | Slightly faster due to additional layers quantized. |
| Aggressive | 262.5 | 439.5 | 0.0 | 1.6 | 0.186 | Fastest configuration; no further accuracy loss. |

## Discussion

All three schedules resulted in identical model sizes due to ONNX Runtime’s INT8-only dynamic quantization, which stores all quantized weights using the same precision format. The cosine drift remained consistent (≈0.19) across schedules because the quantized tensors were nearly identical, reflecting stable embedding alignment following the Phase 3 preprocessing. The Aggressive schedule achieved the lowest latency (−3%) without additional alignment loss, making it the preferred configuration for CPU inference.

## Evaluation Environment

• Python: 3.12.12  
• Platform: Linux 6.6 (x86\_64)  
• Processor: x86\_64  
• Logical cores: 2  
• Physical cores: 1  
• PyTorch: 2.8.0 + cu126  
• NumPy: 1.26.4

## Deliverables

• results\_phase4.csv – Quantization results for all schedules  
• phase4\_cpu\_info.csv – Evaluation environment details  
• phase4\_pareto\_size\_recall.png – Size vs Recall@1 Pareto curve  
• phase4\_latency\_recall.png – Latency vs Recall@1 Pareto curve

## Conclusion

Phase 4 successfully demonstrated mixed-precision PTQ on CPU using ONNX Runtime’s dynamic quantization. While all models used INT8 precision due to framework limitations, selective quantization showed measurable latency improvements with stable alignment and accuracy. The Aggressive schedule achieved the best trade-off between speed and accuracy and is designated as the champion configuration for CPU deployment. Future work in Phase 5 will explore sub-8-bit (INT6/INT4) quantization using Intel Neural Compressor and hybrid calibration methods for further efficiency gains.