This section presents practical details of the evaluation process and how analysis will be conducted.

0.1 Preliminary considerations

Environment - To simulate an edge compute environment we will use the Intel Neural Compute stick for inference, training and compression will be performed on a consumer grade GPU, these environment choices satisfy Aim 3.

Determine models, and test datasets - In accordance with objectives O1 & O2, at least 1 popular model will be selected, the models layer structure should be considered during selection: depth, number of parameters, and number of convolutionalFC layers will be taken into account. Two datasets will be used with these models, one with a small number of classes such as CIFAR-10 and also a dataset with a much larger number of classes such as Imagenet. Ideally the selected model should have pretrained weights for both datasets, if necessary we will train and store the models ourselves.

Select compression algorithms - Select at least 4 algorithms that utilise different pruning methods. If feasable these algorithms should explore a spectrum of sub-domains of Pruning (such as fine-grained pruning vs filter or channel pruning). Any selected algorithms should have the capability to be applied to a specific layer (for this reason knowledge distillation techniques would not be suitable here). For compatibility reasons with the ONNX intermediate representation quantisation may not be supported yet, *Experiment stage 0* (Section 0.2.1) will present a good opportunity to investigate this. The selection of compression algorithms will also depend on how they are implemented within the Intel Distiller framework, for example Automated Gradual Pruner works on a diverse set of neural network architectures so it would be a suitable choice [1].

0.2 Experiment tasks

0.2.1 Experiment stage 0: Verify preliminary results

This experimental stage is necessary due to the preliminary evaluation reported in Section **??**.

The following steps will be taken to complete objective O0:

1. **Evaluate Compression Scheduler**: We will begin by closely assessing the compression scheduler behaviour, there are extensive tools for evaluating the sparsity metrics of pruned models. This will give us insight into how the compression algorithm is affecting the model

2. **Evaluate Intermediate Representation**: We used the ONNX format (an open standard format for represnting machinea learning models), to transfer our model from distiller to OpenVINO. We will take a closer look at this representation for issues with compatibility of sparse tensorsquantisation and the conversion process. One quick verification strategy is to convert the ONNX represtation back to distiller and re-evaluate the models sparsity properties

3. **Evaluate OpenVINO Representations**: Models converted from ONNX format by OpenVINO are then translated into a format consumable by OpenVINO's Inference Engine, this conversion process uses an OpenVINO tool called the Model Optimser. Transformations on the model during these stages will need to be investigated to confirm they do not interfere with the compressed model.

0.2.2 Experiment stage 1: Initial data gathering

Completion of the following stages will satisfy objectives O3, O4, and O5

1. **Aqcuire suite of baseline data**: Using a fixed test set from each dataset we will run inference on all the models with no compression techniques applied, to acquire a *baseline*. The end to end latency, individual layer latency and also the Top1/Top5 accuracy will be recored for each model/dataset pairing.

2. **Apply compression and gather full compression data**: For each compression algorithm and preselected parameters compress the models used in the *baseline* tests by selectively

applying the compression technique to a subset of relevant layers (i.e. layers which the algorithm can be applied). Next using the same testing data from *baseline*, perform inference with the compressed models. The same metrics will be logged as in the *baseline*. We will refer to this test as *full compression*.

3. **Evaluate full compression**: We will make observations about the resulting data, the key metric we are interested in is latency at this stage. First we will make general comparisons with the end-to-end latency and accuracy against the *baseline*. Next we will take a close look at the layer by layer latency against the *baseline*, to try and identify patterns with respect to the size and type of each layer, its location in the neural network, and varience in latency.

4. **Apply combinations of compression techniques**: Based on the results in the previous step we will cherry pick the best agorithm/parameter pairings, with respect to latency reduction for each domain represented in the selected algorithms. We will then apply a composition of these sucessful compression techinques to the models, using the same compression application strategies from *full compression*.

5. **Evaluate combined compression**: We will evaluate latency changes from *full compression* and *baseline*. Of particular interest will be any changes in the individual layer latencies.

0.2.3 Experiment stage 2: Develop optimisation framework

These stages concern objectives O6 and O7

1. **Parameterise compression algorithms**: Develop an interface to define the compression algorithm and its (distiller) scheduler settings. This will be a thin layer on top of distillers pre-existing scheduler api, the purpose of which will be to facilitate communication between an external parameter optimisation tool and distiller.

2. **Implement interface**: We will select the most performant aglorithms from Experiment stage 1 and include select parameters in the aforementioned interface. The parameters selection criteria will be based on observed layerwise latency improvement from Experiment stage 1.

3. **Define optimisation metric**: We will define an optimisation metric using an accuracy threshold as a user defined parameter. This will be the optimisation target.

4. **Integrate interface with benchmark suite**: link optimised distiller model generated via the interface with OpenVINO to run benchmarks

0.2.4 Experiment stage 3: Testing Compression optimisation

This stage of research will complete objective O8

1. **Run the optimiser**: Using the framework developed in stage 2 we will utilise a bayseian search strategy with random forest to identify parameter importance and correlation with the optimisation metric. This will show us which compression parameters are the most important with respect the the metric, and in what direction to tweak them to find an appropriate set of compression parameter values that will result in faster inference within a minimal accuracy threshold.

2.