External inference runtime

inference_wrapper.sh
(bash)

weights file
(model)

assets dir
(code/config)

/cvmfs
(bind if needed)

Apptainer
container

executed in container

--weights

path resolve + bind

bind (optional)

Scratch I/O

planes.bin
(CHW float32)
U,V,W

results.bin
(IAOK header + scores)

metrics.meta
key=value lines

stdout / stderr
(log capture)

--in planes.bin

--metrics

redirect stdout/stderr

parse perf

read scores

Job launcher
lar -c <job>.fcl

selects workflow

ScratchDir
(+ assets base)

FHiCL pset
InferenceProducer

Models[]
Label/Arch
Weights/Wrapper
Assets

resolve paths

PlanesTag,
ScratchDir,
DefaultWrapper

per-model config

write request

Inference stage

InferenceProducer
(EDProducer)

InferenceProduction
runInference(...)

call per model

Result{scores, perf}

produces

Skip inference and write empty products if:
- < 3 planes, or missing U/V/W
- incompatible (W×H) vs adc.size()
- all planes empty (all ADC == 0)

Executes (schematic):
apptainer exec ~~cleanenv~~ bind <assets>,<scratch>[,/cvmfs]\n
  <container> /bin/bash <wrapper>\n
~~in planes.bin~~ out results.bin --metrics metrics.meta\n
~~W <width>~~ H <height> ~~arch <arch>~~ weights <weights>

Then reads results.bin (IAOK/v1) + metrics.meta,\n
and cleans up scratch temp files.

Event products

ImageProducts \"NuSlice\"
U/V/W ADC planes

InferencePredProduct
per-model scores

InferencePerfProduct
per-model timing/RSS

produces

fill per_model

fill per_model