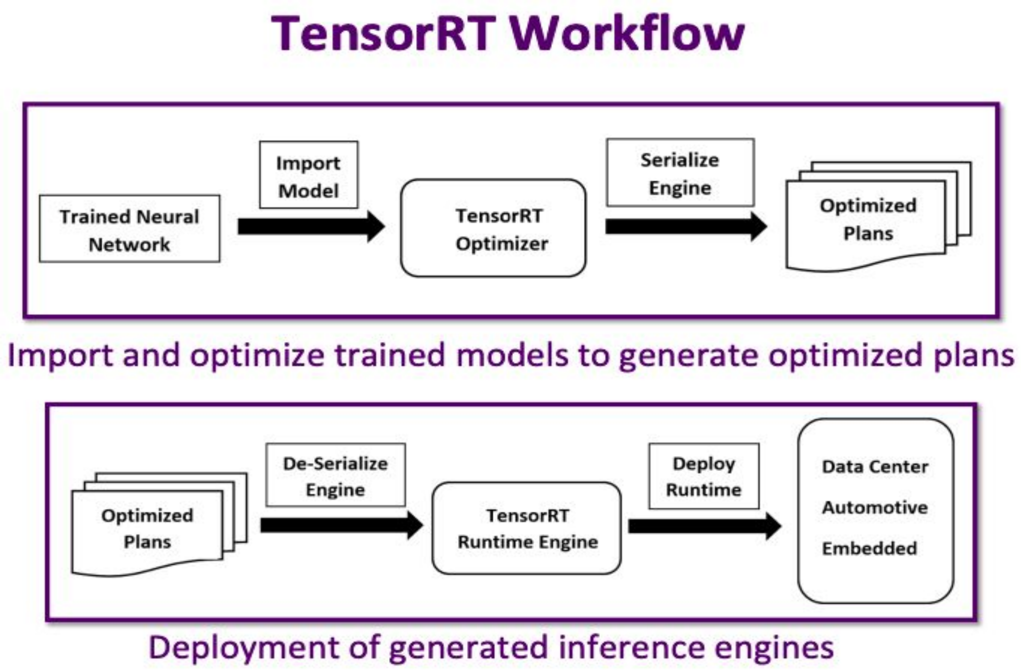
AD Forum Meeting: mpp group, CERN

Syed Anwar Ul Hasan
(Postdoc fellow: Scuola Normale Superiore di Pisa)

9th December, 2021

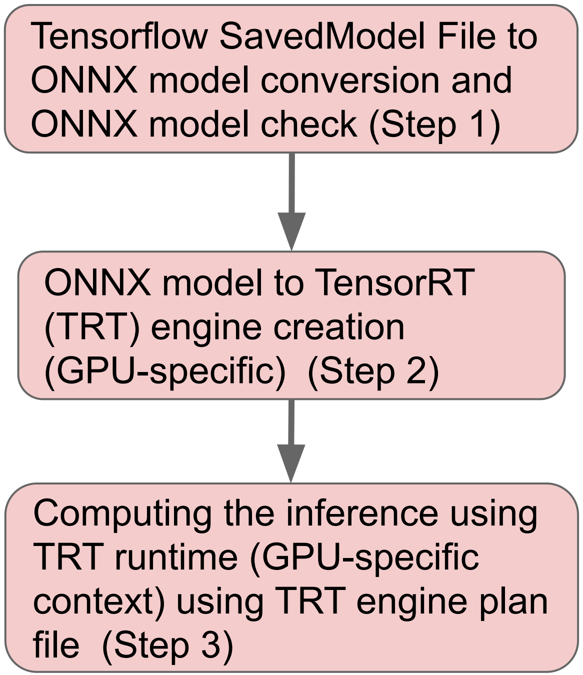
ONNX to TensorRT GPU model inference process



TensorRT Workflow

Import and optimize trained models to generate optimized plans

Deployment of generated inference engines



Tensorflow SavedModel File to ONNX model conversion and ONNX model check (Step 1)

ONNX model to TensorRT (TRT) engine creation (GPU-specific) (Step 2)

Computing the inference using TRT runtime (GPU-specific context) using TRT engine plan file (Step 3)

Current working set-up at mpp-tatooine: TensorRT, CUDA, CUDNN installation

- Instantiating LCG CUDA 101 software stack by running the following command:

  ```
  source /cvmfs/sft.cern.ch/lcg/views/LCG_101cuda/x86_64-centos7-gcc8-opt/setup.sh
  ```

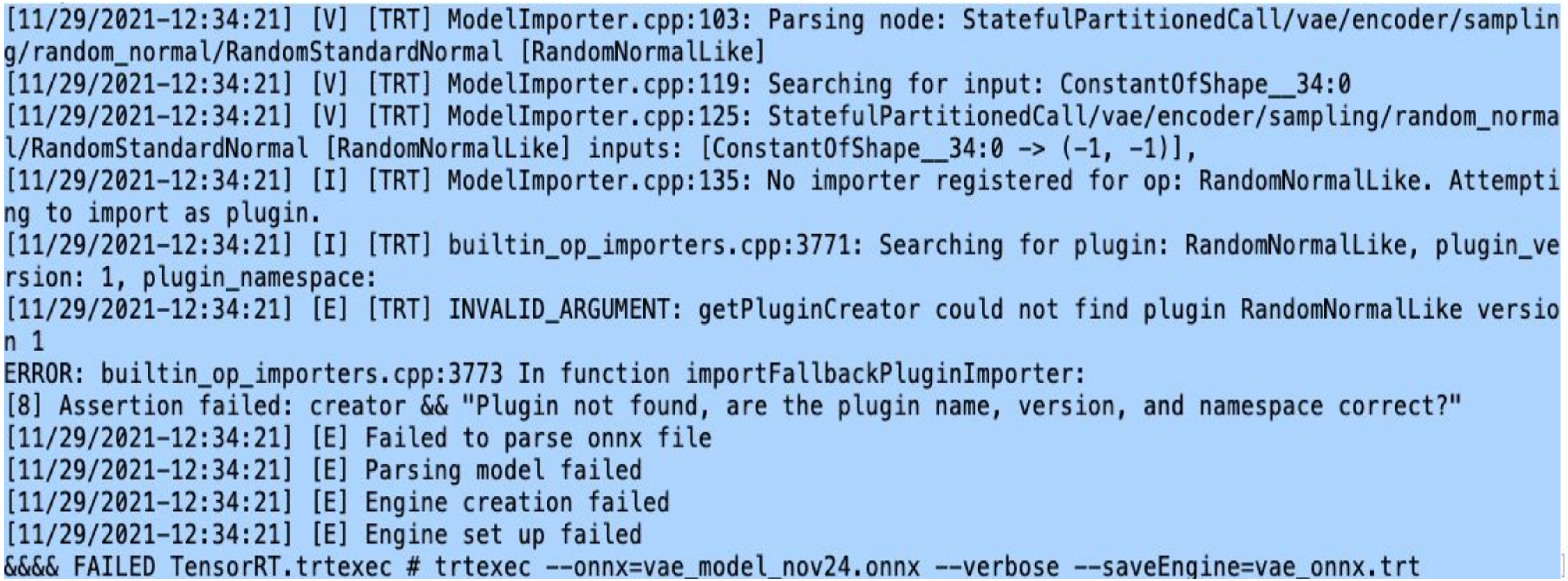
- **Installed versions:**
  - TensorRT 7.2.3.4
  - CUDA 11.2
  - CUDNN 8.1.1.33
  - Tensorflow 2.5.0
  - Onnxruntime 1.8.0
  - Pycuda 2021.1

ONNX to TensorRT model workflow: Tensorflow SavedModel File to ONNX model conversion (Step 1)

- Converting the VAE SavedModel (savedmodel.pb) to ONNX model using **tf2onnx** library
  - With this generated ONNX model, we didn’t succeed (got errors) in creating TRT inference engine file when running TensorRT trtexec tool in step 2.
  - **2 Errors encountered one after another:**
    - RandomNormal distribution (used for epsilon calculation in Sampling layer for reparameterization) not supported by TensorRT
    - ELU activation type not supported by TensorRT
    - Because of this 2 errors, TRT inference engine wasn’t created and thus trtexec didn’t execute completely

ONNX to TensorRT model workflow: Tensorflow SavedModel File to ONNX model conversion (Step 1)

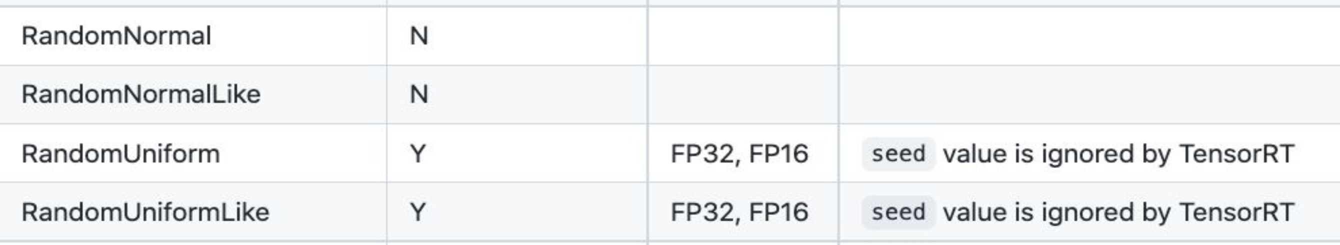
- Error 1: RandomNormal distribution not supported by TensorRT (shown in the Figure below)



```
[11/29/2021-12:34:21] [V] [TRT] ModelImporter.cpp:103: Parsing node: StatefulPartitionedCall/vae/encoder/samplin
g/random_normal/RandomStandardNormal [RandomNormalLike]
[11/29/2021-12:34:21] [V] [TRT] ModelImporter.cpp:119: Searching for input: ConstantOfShape__34:0
[11/29/2021-12:34:21] [V] [TRT] ModelImporter.cpp:125: StatefulPartitionedCall/vae/encoder/sampling/random_norma
l/RandomStandardNormal [RandomNormalLike] inputs: [ConstantOfShape__34:0 -> (-1, -1)],
[11/29/2021-12:34:21] [I] [TRT] ModelImporter.cpp:135: No importer registered for op: RandomNormalLike. Attempti
ng to import as plugin.
[11/29/2021-12:34:21] [I] [TRT] builtin_op_importers.cpp:3771: Searching for plugin: RandomNormalLike, plugin_ve
rsion: 1, plugin_namespace:
[11/29/2021-12:34:21] [E] [TRT] INVALID_ARGUMENT: getPluginCreator could not find plugin RandomNormalLike versio
n 1
ERROR: builtin_op_importers.cpp:3773 In function importFallbackPluginImporter:
[8] Assertion failed: creator && "Plugin not found, are the plugin name, version, and namespace correct?"
[11/29/2021-12:34:21] [E] Failed to parse onnx file
[11/29/2021-12:34:21] [E] Parsing model failed
[11/29/2021-12:34:21] [E] Engine creation failed
[11/29/2021-12:34:21] [E] Engine set up failed
&&&& FAILED TensorRT.trtexec # trtexec --onnx=vae_model_nov24.onnx --verbose --saveEngine=vae_onnx.trt
```

ONNX to TensorRT model workflow: Tensorflow SavedModel File to ONNX model conversion (Step 1)

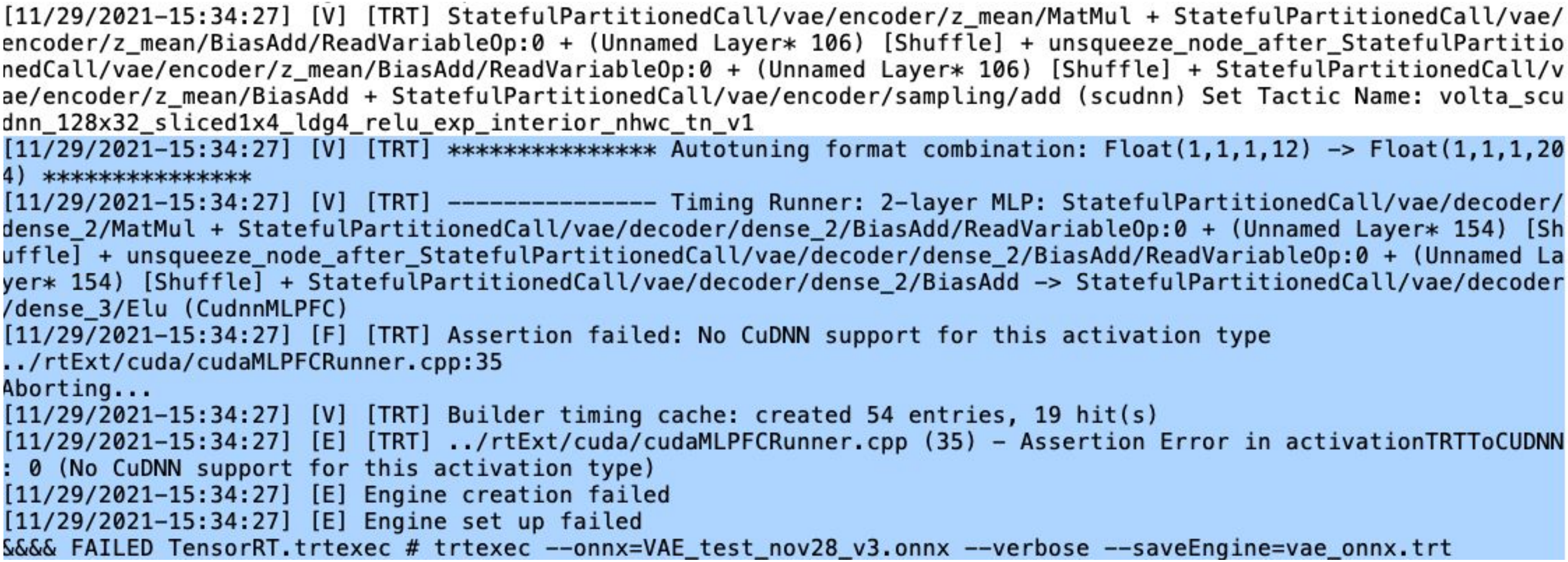
- Methods to resolve Error 1: RandomNormal distribution not supported by TensorRT
  - Employed a different RandomNormal function (tf.random.normal) instead of keras function but couldn't succeed
  - Writing a custom plugin in TensorRT for this function may require dependencies with underlying TensorRT cpp libraries. Didn't get much into the details
  - Found out RandomUniform distribution is supported, so chose epsilon following a random uniform instead of random normal distribution. Need to check further on how VAE behaves with this change.



RandomNormal	N		
RandomNormalLike	N		
RandomUniform	Y	FP32, FP16	seed value is ignored by TensorRT
RandomUniformLike	Y	FP32, FP16	seed value is ignored by TensorRT

ONNX to TensorRT model workflow: Tensorflow SavedModel File to ONNX model conversion (Step 1)

- Error 2: ELU activation type not supported by TensorRT (shown in the Figure below)



```
[11/29/2021-15:34:27] [V] [TRT] StatefulPartitionedCall/vae/encoder/z_mean/MatMul + StatefulPartitionedCall/vae/
encoder/z_mean/BiasAdd/ReadVariableOp:0 + (Unnamed Layer* 106) [Shuffle] + unsqueeze_node_after_StatefulPartitio
nedCall/vae/encoder/z_mean/BiasAdd/ReadVariableOp:0 + (Unnamed Layer* 106) [Shuffle] + StatefulPartitionedCall/v
ae/encoder/z_mean/BiasAdd + StatefulPartitionedCall/vae/encoder/sampling/add (scudnn) Set Tactic Name: volta_scu
dnn_128x32_sliced1x4_ldg4_relu_exp_interior_nhwc_tn_v1
[11/29/2021-15:34:27] [V] [TRT] *************** Autotuning format combination: Float(1,1,1,12) -> Float(1,1,1,20
4) ***************
[11/29/2021-15:34:27] [V] [TRT] --------------- Timing Runner: 2-layer MLP: StatefulPartitionedCall/vae/decoder/
dense_2/MatMul + StatefulPartitionedCall/vae/decoder/dense_2/BiasAdd/ReadVariableOp:0 + (Unnamed Layer* 154) [Sh
uffle] + unsqueeze_node_after_StatefulPartitionedCall/vae/decoder/dense_2/BiasAdd/ReadVariableOp:0 + (Unnamed La
yer* 154) [Shuffle] + StatefulPartitionedCall/vae/decoder/dense_2/BiasAdd -> StatefulPartitionedCall/vae/decoder
/dense_3/Elu (CudnnMLPFC)
[11/29/2021-15:34:27] [F] [TRT] Assertion failed: No CuDNN support for this activation type
../rtExt/cuda/cudaMLPFCRunner.cpp:35
Aborting...
[11/29/2021-15:34:27] [V] [TRT] Builder timing cache: created 54 entries, 19 hit(s)
[11/29/2021-15:34:27] [E] [TRT] ../rtExt/cuda/cudaMLPFCRunner.cpp (35) - Assertion Error in activationTRTToCUDNN
: 0 (No CuDNN support for this activation type)
[11/29/2021-15:34:27] [E] Engine creation failed
[11/29/2021-15:34:27] [E] Engine set up failed
&&&& FAILED TensorRT.trtexec # trtexec --onnx=VAE_test_nov28_v3.onnx --verbose --saveEngine=vae_onnx.trt
```

ONNX to TensorRT model workflow: Tensorflow SavedModel File to ONNX model conversion (Step 1)

- Method to resolve the Error 2: ELU activation type not supported by TensorRT
    - Instead of ELU, I chose ReLU activation type for the VAE model during training

- With RandomUniform distribution for epsilon and ReLU activation type, the generated ONNX model file is compatible with TensorRT trtexec tool and the TRT engine file (plan) is created.

- We generated separate TRT engine plans for Tesla V100 and Tesla T4 GPU, and also created separate TRT context for each of them for the inference run.

ONNX to TensorRT model workflow: ONNX model to TensorRT (TRT) TRTExec tool for TRT engine creation (Step 2)

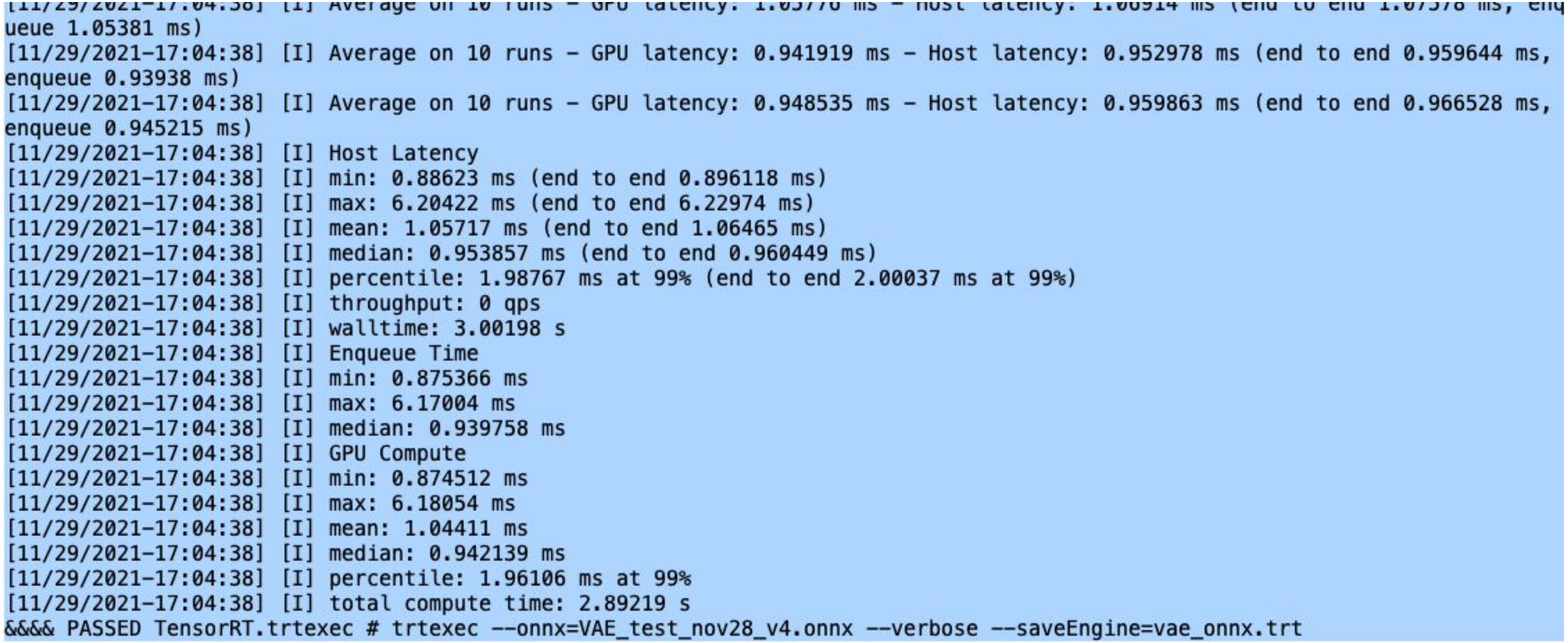
- Using TensorRT in-built trtexec tool, we create a TensorRT engine file from the ONNX model (the text file log of operations trtexec generates is very big - tens of pages)

- Since TensorRT and also trtexec works with CUDA, CUDDN, Pycuda working set-up of each one of them is required to generate the TRT engine file.

- Run: TRT_EXEC --onnx=onnx_model_name --output=trt_engine.trt

ONNX to TensorRT model workflow: ONNX model to TensorRT (TRT) TRTExec tool for TRT engine creation (Step 2)

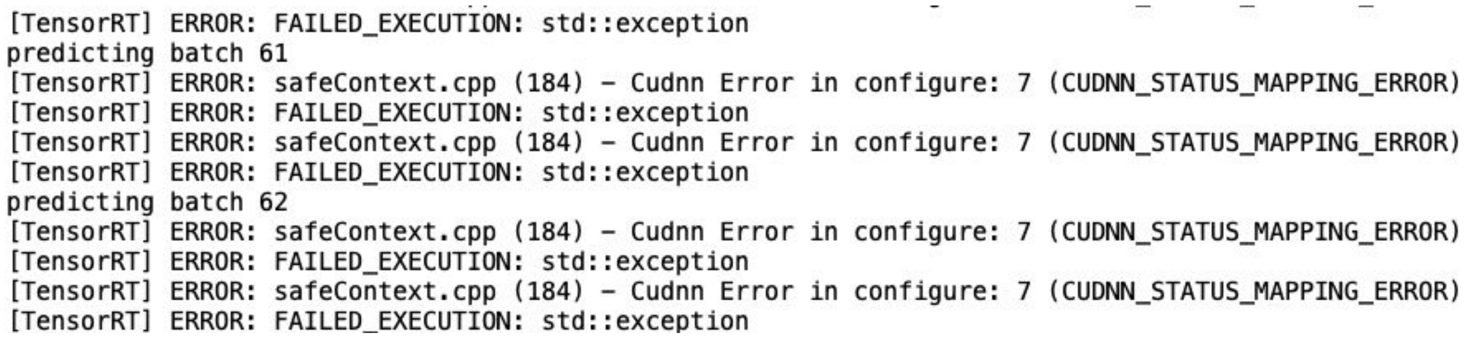
- TRTexec is successful (shows the PASSED message at the end of the run) and generates TRT engine file for the VAE onnx model.



```
[11/29/2021-17:04:38] [I] Average on 10 runs - GPU latency: 1.05776 ms - Host latency: 1.06914 ms (end to end 1.07378 ms, enq
ueue 1.05381 ms)
[11/29/2021-17:04:38] [I] Average on 10 runs - GPU latency: 0.941919 ms - Host latency: 0.952978 ms (end to end 0.959644 ms,
enqueue 0.93938 ms)
[11/29/2021-17:04:38] [I] Average on 10 runs - GPU latency: 0.948535 ms - Host latency: 0.959863 ms (end to end 0.966528 ms,
enqueue 0.945215 ms)
[11/29/2021-17:04:38] [I] Host Latency
[11/29/2021-17:04:38] [I] min: 0.88623 ms (end to end 0.896118 ms)
[11/29/2021-17:04:38] [I] max: 6.20422 ms (end to end 6.22974 ms)
[11/29/2021-17:04:38] [I] mean: 1.05717 ms (end to end 1.06465 ms)
[11/29/2021-17:04:38] [I] median: 0.953857 ms (end to end 0.960449 ms)
[11/29/2021-17:04:38] [I] percentile: 1.98767 ms at 99% (end to end 2.00037 ms at 99%)
[11/29/2021-17:04:38] [I] throughput: 0 qps
[11/29/2021-17:04:38] [I] walltime: 3.00198 s
[11/29/2021-17:04:38] [I] Enqueue Time
[11/29/2021-17:04:38] [I] min: 0.875366 ms
[11/29/2021-17:04:38] [I] max: 6.17004 ms
[11/29/2021-17:04:38] [I] median: 0.939758 ms
[11/29/2021-17:04:38] [I] GPU Compute
[11/29/2021-17:04:38] [I] min: 0.874512 ms
[11/29/2021-17:04:38] [I] max: 6.18054 ms
[11/29/2021-17:04:38] [I] mean: 1.04411 ms
[11/29/2021-17:04:38] [I] median: 0.942139 ms
[11/29/2021-17:04:38] [I] percentile: 1.96106 ms at 99%
[11/29/2021-17:04:38] [I] total compute time: 2.89219 s
&&&& PASSED TensorRT.trtexec # trtexec --onnx=VAE_test_nov28_v4.onnx --verbose --saveEngine=vae_onnx.trt
```

ONNX to TensorRT model workflow: Computing the inference with TRT runtime with TRT engine file as input (Step 3)

- Currently, I am getting CUDDN mapping error when I run the TRT engine file with the TensorRT runtime



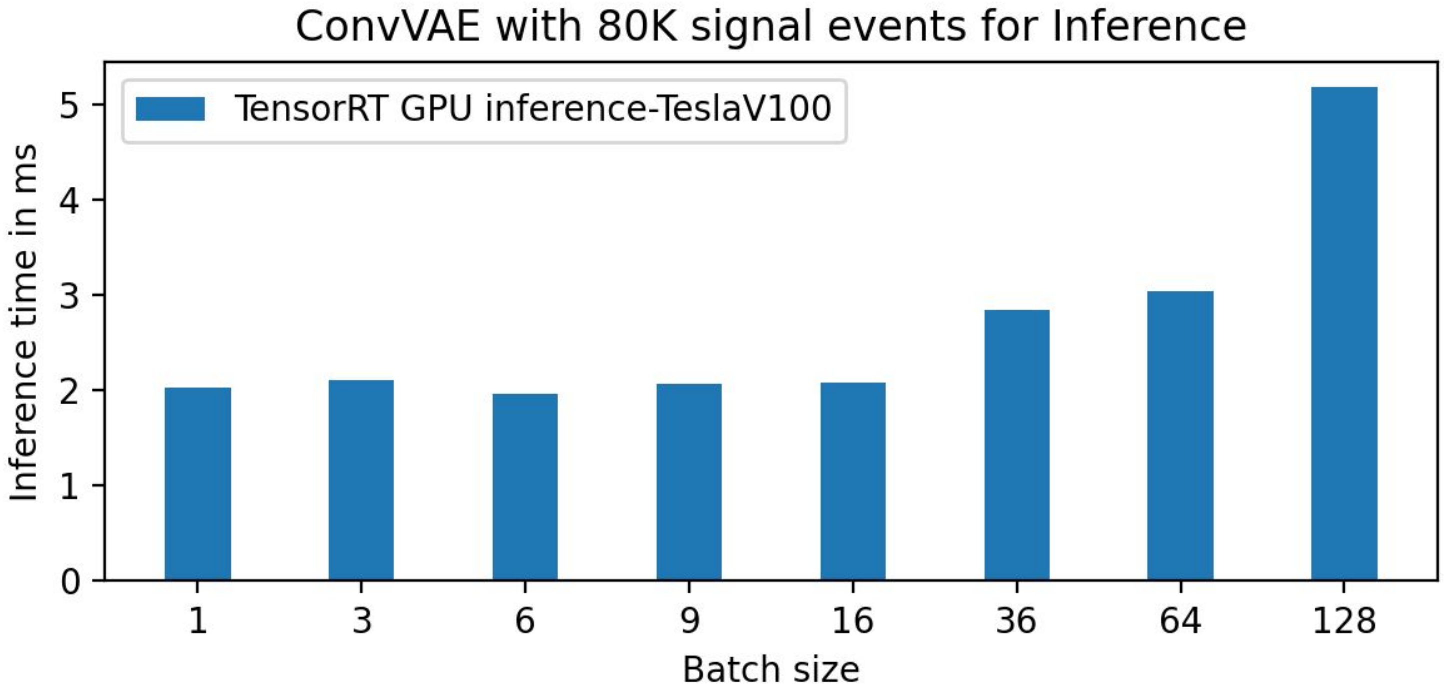
```
[TensorRT] ERROR: FAILED_EXECUTION: std::exception
predicting batch 61
[TensorRT] ERROR: safeContext.cpp (184) - Cudnn Error in configure: 7 (CUDNN_STATUS_MAPPING_ERROR)
[TensorRT] ERROR: FAILED_EXECUTION: std::exception
[TensorRT] ERROR: safeContext.cpp (184) - Cudnn Error in configure: 7 (CUDNN_STATUS_MAPPING_ERROR)
[TensorRT] ERROR: FAILED_EXECUTION: std::exception
predicting batch 62
[TensorRT] ERROR: safeContext.cpp (184) - Cudnn Error in configure: 7 (CUDNN_STATUS_MAPPING_ERROR)
[TensorRT] ERROR: FAILED_EXECUTION: std::exception
[TensorRT] ERROR: safeContext.cpp (184) - Cudnn Error in configure: 7 (CUDNN_STATUS_MAPPING_ERROR)
[TensorRT] ERROR: FAILED_EXECUTION: std::exception
```

- TRT model is getting created from the engine file but error in the inference runtime phase before computing the predictions.

- I solved this Pycuda error by using push and pop methods and deleting the context after inference run

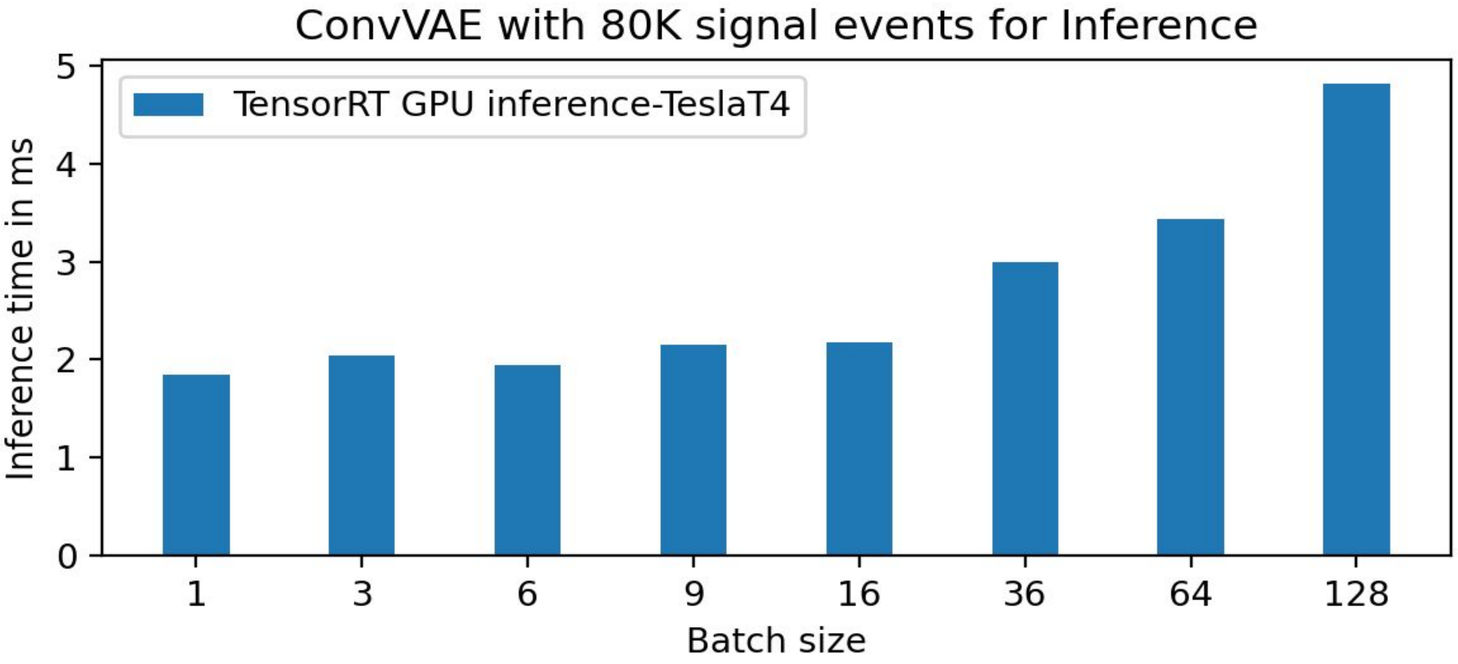
ONNX to TensorRT model workflow: RESULTS after computing the inference with TRT runtime using TRT engine (plans)

- **GPU: TESLA V100 (specific TRT engine plan and context), Precision: FP32**



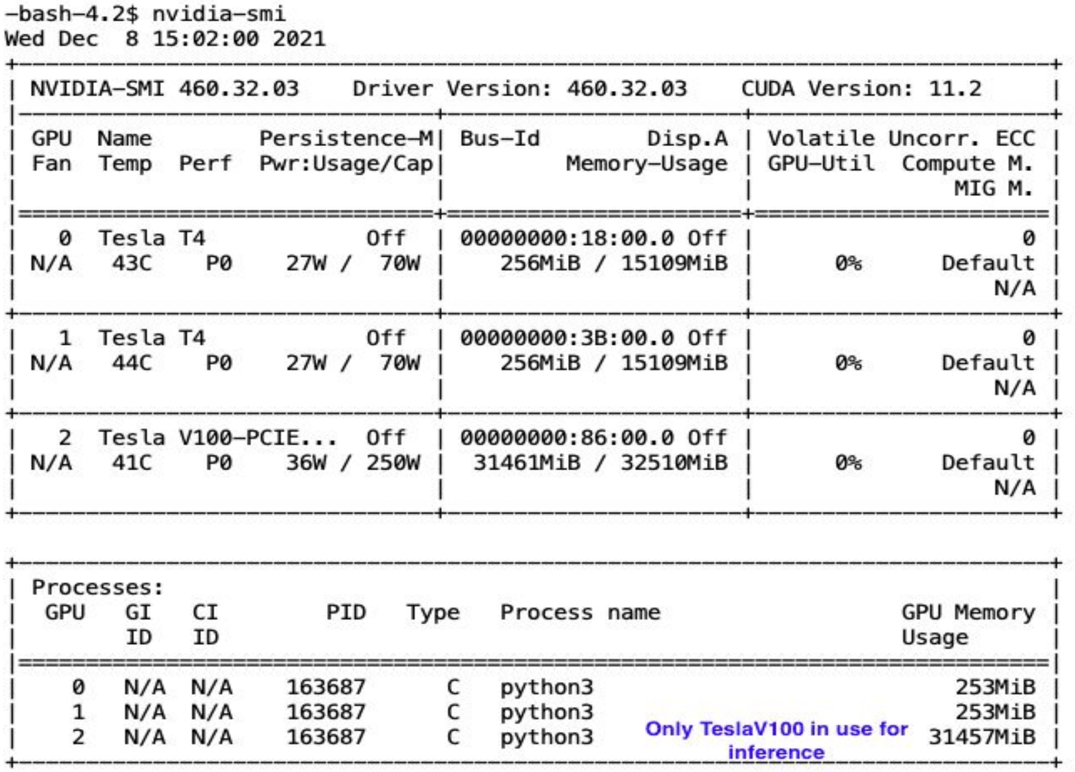
ONNX to TensorRT model workflow: RESULTS after computing the inference with TRT runtime using TRT engine (plans)

- **GPU: TESLA T4 (specific TRT engine plan and context), Precision: FP32**



ONNX to TensorRT model workflow: RESULTS after computing the inference with TRT runtime using TRT engine (plans)

- **TESLA V100**



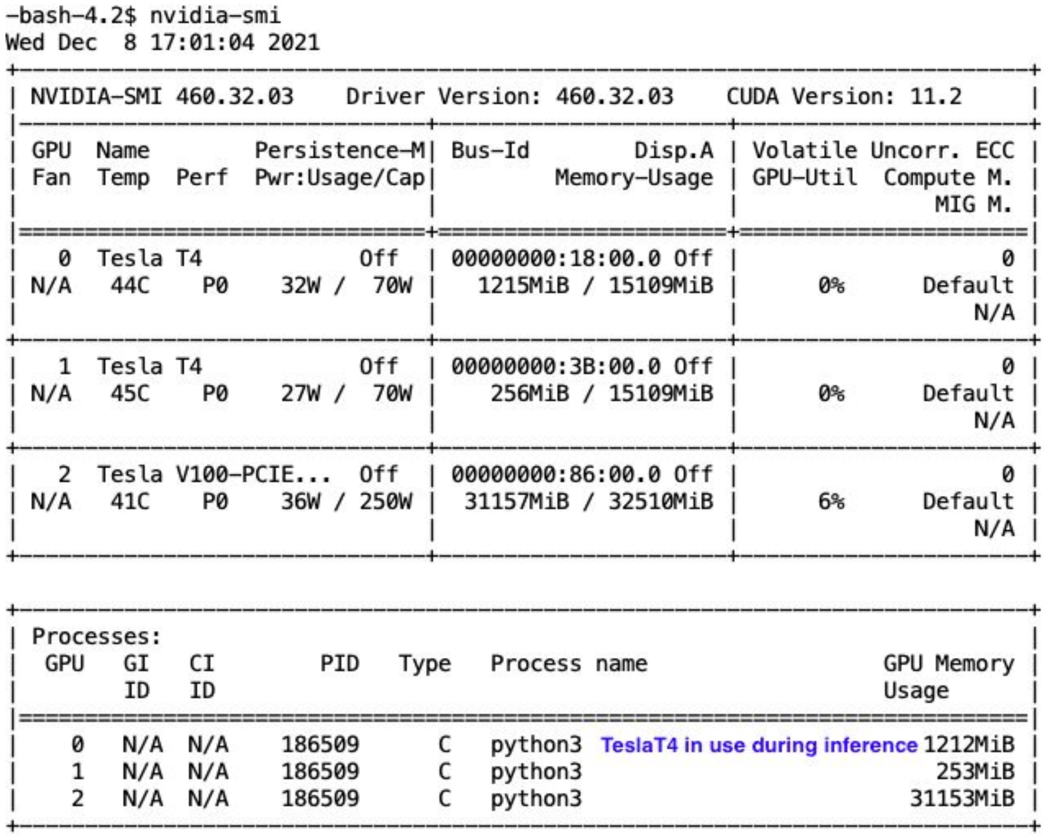
```
-bash-4.2$ nvidia-smi
Wed Dec  8 15:02:00 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:18:00.0 Off |                    0 |
| N/A   43C    P0    27W /  70W |    256MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  Tesla T4            Off  | 00000000:3B:00.0 Off |                    0 |
| N/A   44C    P0    27W /  70W |    256MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   2  Tesla V100-PCIE...  Off  | 00000000:86:00.0 Off |                    0 |
| N/A   41C    P0    36W / 250W |  31461MiB / 32510MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A    163687      C   python3                           253MiB |
|    1   N/A  N/A    163687      C   python3                           253MiB |
|    2   N/A  N/A    163687      C   python3                         31457MiB |
+-----------------------------------------------------------------------------+
```

Only TeslaV100 in use for inference

ONNX to TensorRT model workflow: RESULTS after computing the inference with TRT runtime using TRT engine (plans)

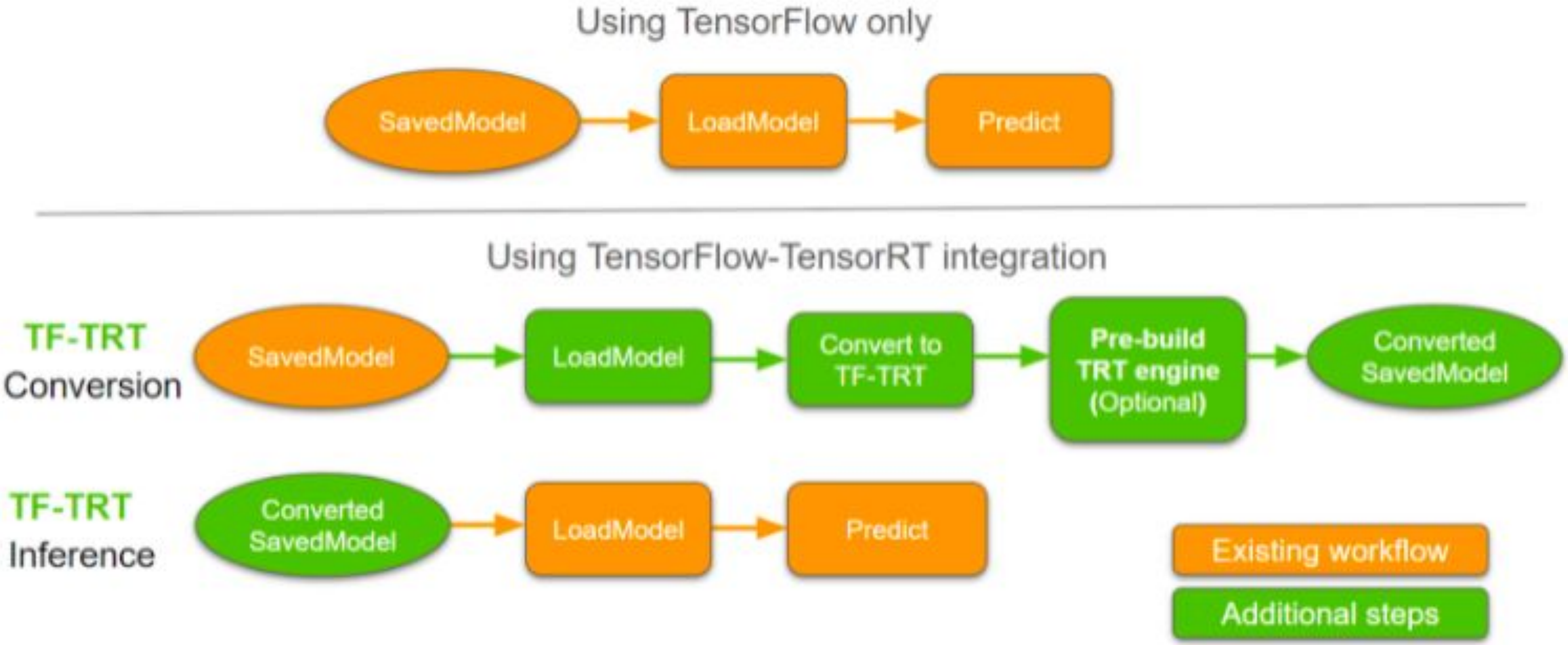
- **TESLA T4**



```
-bash-4.2$ nvidia-smi
Wed Dec  8 17:01:04 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:18:00.0 Off |                    0 |
| N/A   44C    P0    32W /  70W |   1215MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  Tesla T4            Off  | 00000000:3B:00.0 Off |                    0 |
| N/A   45C    P0    27W /  70W |    256MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   2  Tesla V100-PCIE...  Off  | 00000000:86:00.0 Off |                    0 |
| N/A   41C    P0    36W / 250W |  31157MiB / 32510MiB |      6%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A    186509      C   python3  TeslaT4 in use during inference 1212MiB |
|    1   N/A  N/A    186509      C   python3                          253MiB |
|    2   N/A  N/A    186509      C   python3                        31153MiB |
+-----------------------------------------------------------------------------+
```

TF-TensorRT model inference (native TF) with GPUs



Using TensorFlow only

SavedModel → LoadModel → Predict

Using TensorFlow-TensorRT integration

TF-TRT Conversion

SavedModel → LoadModel → Convert to TF-TRT → Pre-build TRT engine (Optional) → Converted SavedModel

TF-TRT Inference

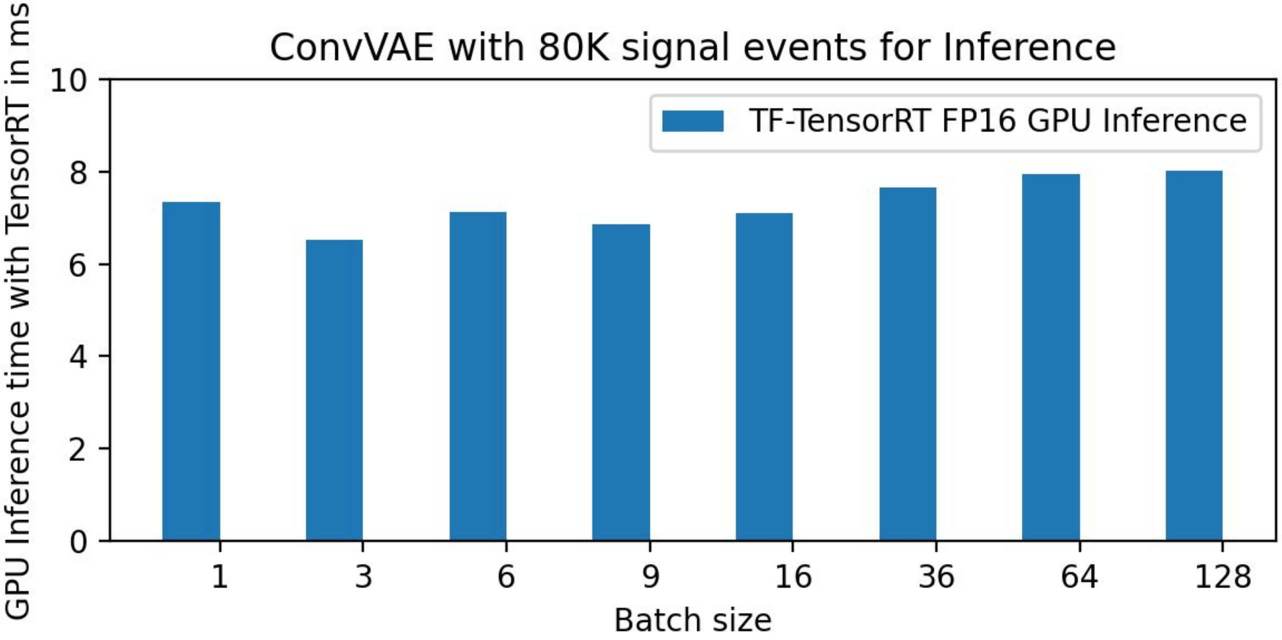
Converted SavedModel → LoadModel → Predict

Existing workflow

Additional steps

TF-TensorRT model inference: Results

- The trend is the inference time stays relatively flat for different batch sizes


ConvVAE with 80K signal events for Inference
GPU Inference time with TensorRT in ms
TF-TensorRT FP16 GPU Inference
Batch size