**3GPP TSG-SA WG4 Meeting #127-bis-eS4-240647\_r01**

**E-meeting, 8 April - 12 April 2024**

**Source: Samsung Electronics Co., Ltd.**

**Title: [FS\_AI4Media] pCR on existing frameworks**

**Spec: 3GPP TR 26.927 v0.6.0**

**Agenda item: 9.6**

**Document for: Agreement**

**1. Introduction**

Includes PyTorch as an existing framework, with text from the PD.

**2. Reason for change**

Only description on TensorFlow exists in existing frameworks clause.

**3. Proposal**

It is proposed to agree the following changes to 3GPP TR 26.927 v0.6.0.

\* \* \* First Change \* \* \* \*

### 6.4.2 PyTorch

#### 6.4.2.1 Introduction

PyTorch is based on the concept of tensors, which are multi-dimensional arrays of numerical data. Similarly to TensorFlow, Tensors are a fundamental data structure used in PyTorch to represent the input data and the parameters of the machine learning model. PyTorch provides a range of operations for manipulating tensors, such as addition, multiplication, and convolution.

PyTorch also supports dynamic computation graphs, which allow for more flexibility in building and training machine learning models. This means that the computational graph can be modified on-the-fly during runtime, which makes it easier to build complex models and experiment with different architectures. Additionally, PyTorch provides a high-level API called TorchScript, which allows for models to be exported to a portable format that can be executed on various platforms.

#### 6.4.2.2 Main differences with TensorFlow

**Computational graph:** TensorFlow uses a static computational graph, which means that the graph is defined and compiled before the training begins. On the other hand, PyTorch uses a dynamic computational graph, which allows for more flexibility in building and modifying the graph during runtime.

**Ease of use:** PyTorch is generally considered to be more user-friendly and simpler than TensorFlow. This is partly due to its dynamic computational graph, which makes it easier to experiment with different models and architectures. PyTorch also has a more Python-like syntax, which is familiar to many developers.

**Visualization:** TensorFlow provides comprehensive visualization tools (such as TensorBoard), which allows users to monitor the training progress and visualize the model's performance. PyTorch does not have a built-in visualization tool, but can be integrated with Tensorboard, and there are also several third-party libraries available, such as Visdom.

**Ecosystem:** TensorFlow has better support for deploying models on mobile devices and in production environments. However, PyTorch has been gaining popularity in recent years and has a growing ecosystem.

**Research:** PyTorch is more popular in the research community, as it allows for faster prototyping and experimentation due to its dynamic computational graph. TensorFlow is more commonly used in industry for production-level applications due to its static graph and better support for deployment.

\* \* \* End of Changes \* \* \* \*