Meeting of the LF AI & Data Technical Advisory Council (TAC)

July 13, 2023

LF AI & DATA

Antitrust Policy

- Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

- Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrust-policy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Undergone LLP, which provides legal counsel to the Linux Foundation.

Recording of Calls

Reminder:

TAC calls are recorded and available for viewing on the TAC Wiki

Reminder: LF AI & Data Useful Links

- Web site: lfaidata.foundation
- Wiki: wiki.lfaidata.foundation
- GitHub: github.com/lfaidata
- Landscape: https://landscape.lfaidata.foundation or https://l.lfaidata.foundation
- Mail Lists: https://lists.lfaidata.foundation
- Slack: https://slack.lfaidata.foundation
- Youtube: https://www.youtube.com/channel/UCfasaeqXJBCAJMNO9HcHfbA
- LF AI Logos: https://github.com/lfaidata/artwork/tree/master/lfaidata
- LF AI Presentation Template: https://drive.google.com/file/d/1eiDNJvXCqSZHT4Zk_-czASlz2GTBRZk2/view?usp=sharing

- Events Page on LF AI Website: https://lfaidata.foundation/events/
- Events Calendar on LF AI Wiki (subscribe available): https://wiki.lfaidata.foundation/pages/viewpage.action?pageId=12091544
- Event Wiki Pages: https://wiki.lfaidata.foundation/display/DL/LF+AI+Data+Foundation+Events

Agenda

- Roll Call (1 mins)
- Approval of Minutes from previous meeting (2 mins)
- ShaderNN (40 minutes)
- Open Discussion

TAC Voting Members - Please note

Please ensure that you do the following to facilitate smooth procedural quorum and voting processes:

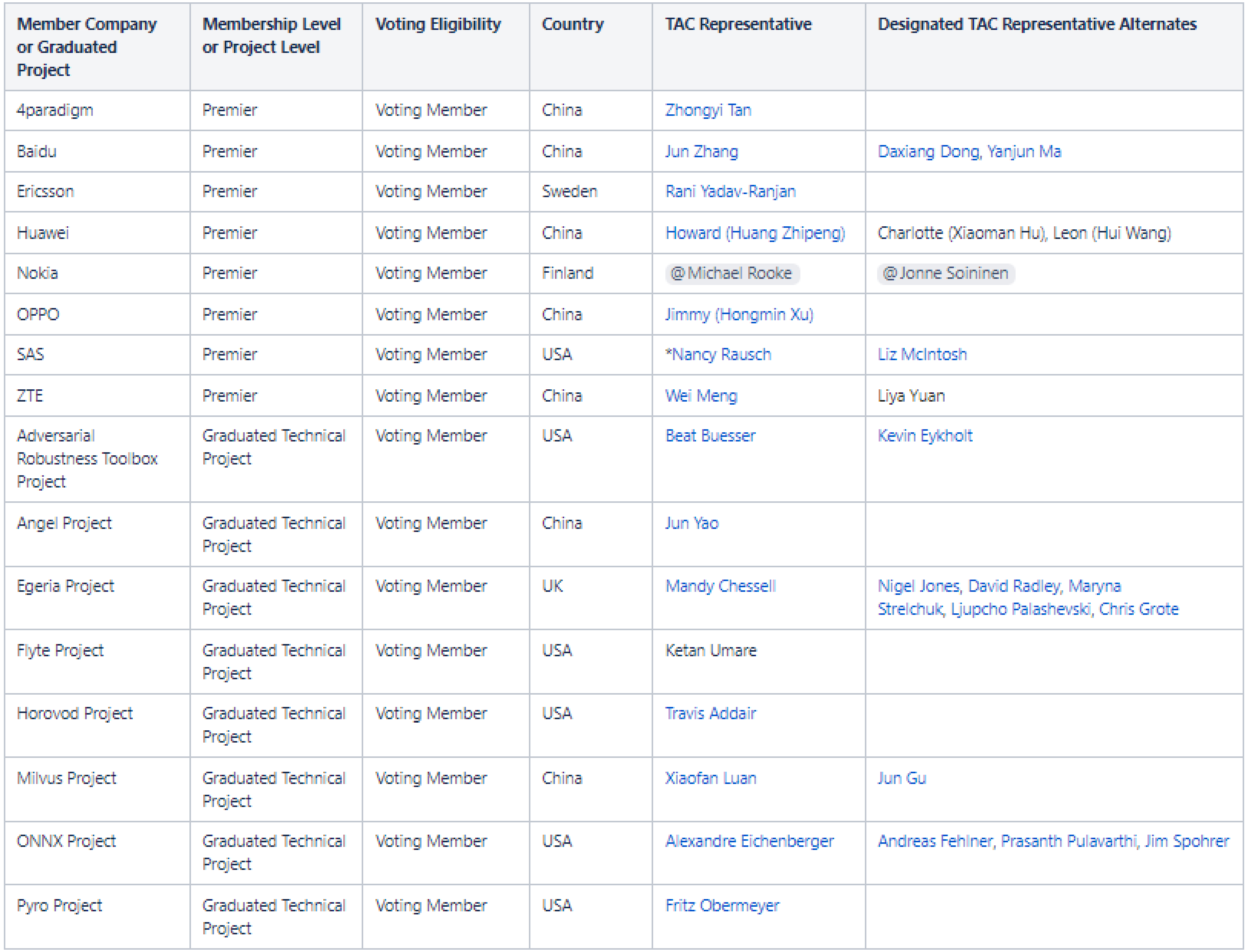
- Change your Zoom display name to include your First/Last Name, Company/Project Represented
  - example: Nancy Rausch, SAS
- State your First/Last Name and Company/Project when submitting a motion
  - example: First motion, Nancy Rausch/SAS

TAC Voting Members - Please note

- TAC members must attend consistently to maintain their voting status
- After 2 absences voting members will lose voting privileges
- Voting privileges will only be reinstated after attending 2 meetings in a row

TAC Voting Members

Note: we still need a few designated backups specified on wiki



Member Company or Graduated Project	Membership Level or Project Level	Voting Eligibility	Country	TAC Representative	Designated TAC Representative Alternates
4paradigm	Premier	Voting Member	China	Zhongyi Tan	
Baidu	Premier	Voting Member	China	Jun Zhang	Daxiang Dong, Yanjun Ma
Ericsson	Premier	Voting Member	Sweden	Rani Yadav-Ranjan	
Huawei	Premier	Voting Member	China	Howard (Huang Zhipeng)	Charlotte (Xiaoman Hu), Leon (Hui Wang)
Nokia	Premier	Voting Member	Finland	@Michael Rooke	@Jonne Soininen
OPPO	Premier	Voting Member	China	Jimmy (Hongmin Xu)	
SAS	Premier	Voting Member	USA	*Nancy Rausch	Liz McIntosh
ZTE	Premier	Voting Member	China	Wei Meng	Liya Yuan
Adversarial Robustness Toolbox Project	Graduated Technical Project	Voting Member	USA	Beat Buesser	Kevin Eykholt
Angel Project	Graduated Technical Project	Voting Member	China	Jun Yao	
Egeria Project	Graduated Technical Project	Voting Member	UK	Mandy Chessell	Nigel Jones, David Radley, Maryna Strelchuk, Ljupcho Palashevski, Chris Grote
Flyte Project	Graduated Technical Project	Voting Member	USA	Ketan Umare	
Horovod Project	Graduated Technical Project	Voting Member	USA	Travis Addair	
Milvus Project	Graduated Technical Project	Voting Member	China	Xiaofan Luan	Jun Gu
ONNX Project	Graduated Technical Project	Voting Member	USA	Alexandre Eichenberger	Andreas Fehlner, Prasanth Pulavarthi, Jim Spohrer
Pyro Project	Graduated Technical Project	Voting Member	USA	Fritz Obermeyer	

Minutes approval

LF AI & DATA

Approval of June 29, 2023 Minutes

Draft minutes from the June 29 TAC call were previously distributed to the TAC members via the mailing list

Proposed Resolution:

- That the minutes of the June 29 meeting of the Technical Advisory Council of the LF AI & Data Foundation are hereby approved.

ShaderNN: A Shader Based Lightweight and Efficient Inference Engine for Mobile GPU

2023/7/13

OPPO Computing & Graphics Research Institute

Agenda

1 Why donate to LF AI & Data

2 Challenges for Mobile Inference

3 What is ShaderNN?

4 ShaderNN Open Source & Roadmap

Why donate to LF AI & Data

Why donate to LF AI & Data

- **Collaborative Development and Community Support**:

  Leverage the collective knowledge, expertise, and resources of the diverse community of developers, researchers, and organizations to advance our project and gain support, feedback, and contributions.

- **Visibility and Exposure**:

  Attract new contributors, users, and supporters by promoting our organization in AI and data communities.

- **Legal and Governance Support**:

  Ensure compliance with relevant laws and regulations and operation in a transparent and fair manner.

- **Long-Term Sustainability**:

  Guarantee continuous maintenance and support by a vibrant and active community for years to come.

Agenda

1 Why donate to LF AI & Data

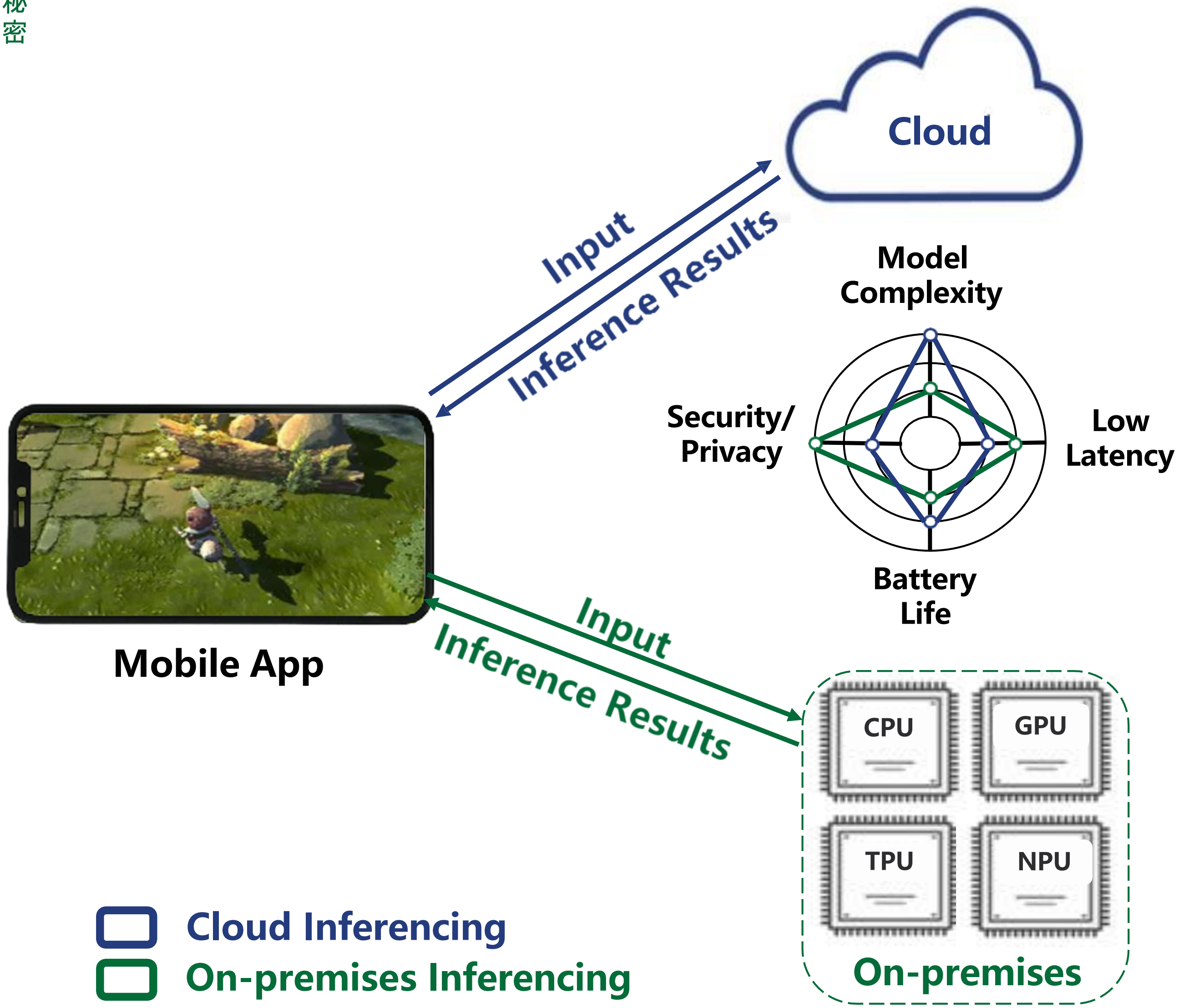
2 Challenges for Mobile Inference

3 What is ShaderNN?

4 ShaderNN Open Source & Roadmap

Mobile Inference Engine Overview

Mobile Inference Engine Overview



Major challenges for on-premises inference:

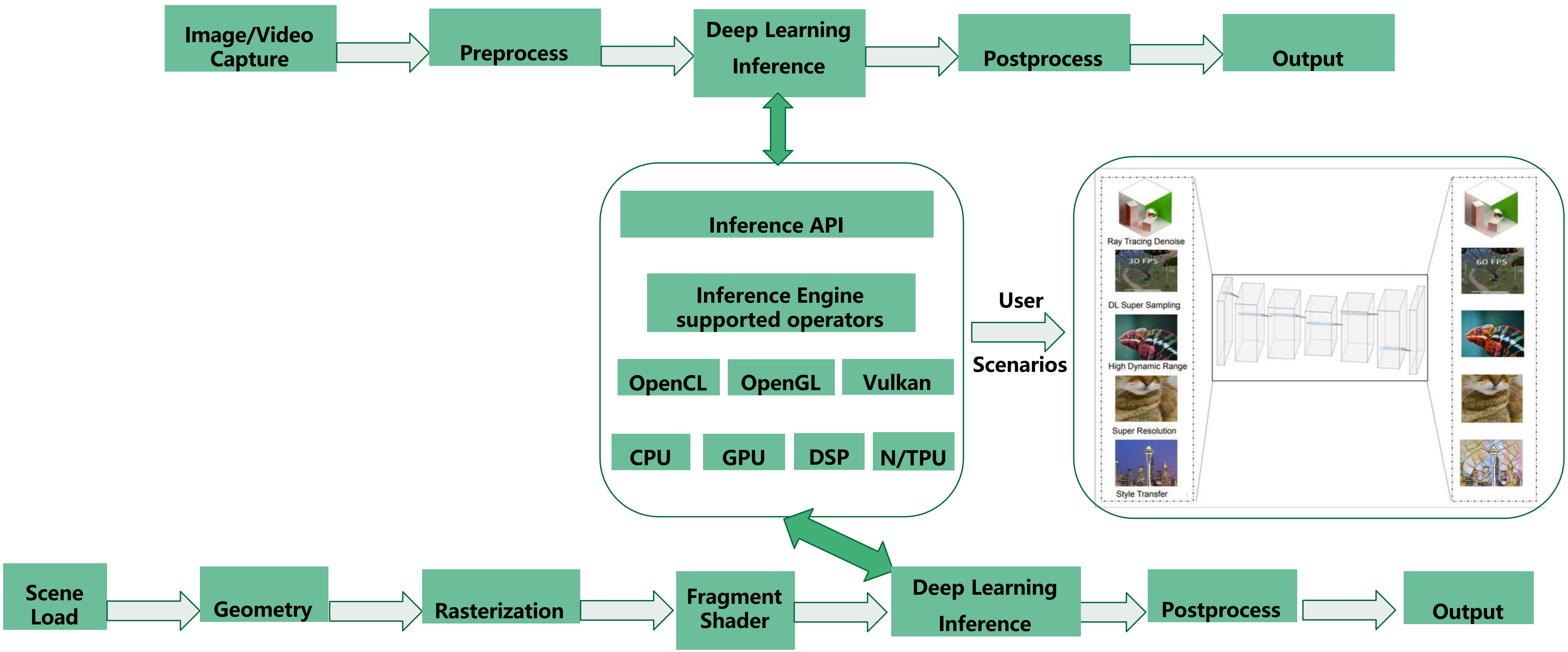
- Limited computational capacity.
- Low power budget.
- Model compatibility.
- Customizable and lightweight implementation.
- Deeply coupled with image/graphic applications.
- Varied memory access methods and I/O bus bandwidth.

	CPU	SIMD	OpenCL	Open GL Compute Shader	Open GL Fragment Shader	Vulkan	NPU/ DSP
TensorFlow Lite	V	V	V	V			V
MNN	V	V	V	V		V	V
NCNN	V	V				V	
TNN	V	V	V				V
BOLT	V	V	V				
MACE	V	V	V				V
ShaderNN	V			V	V	V	

Challenges for Image/Video/Graphics AI applications

Challenges for Image/Video/Graphics AI applications

Image/Video Applications



Image/Video Capture → Preprocess → Deep Learning Inference → Postprocess → Output

Inference API

Inference Engine supported operators

OpenCL | OpenGL | Vulkan

CPU | GPU | DSP | N/TPU

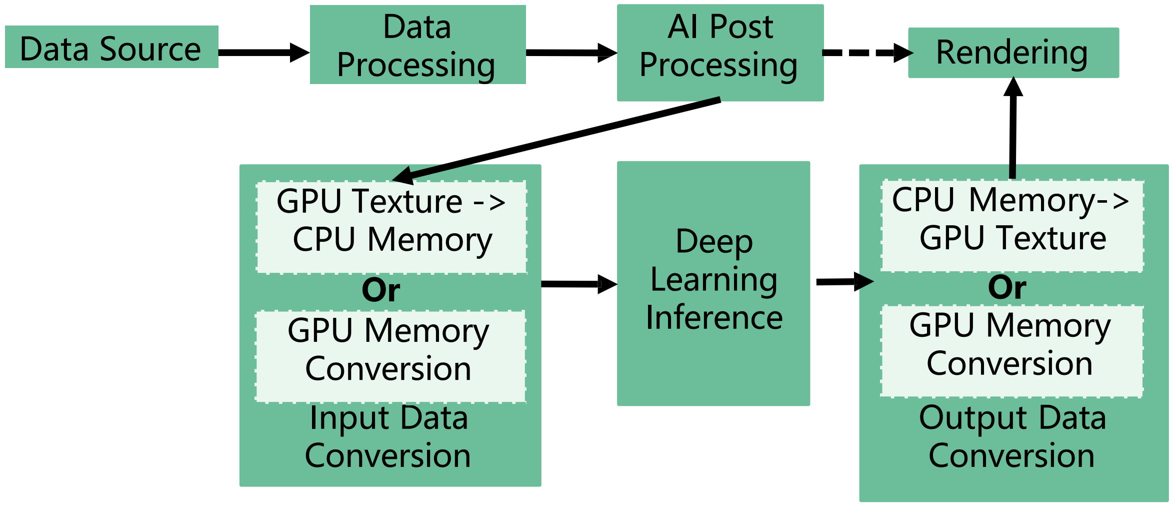
User Scenarios

Scene Load → Geometry → Rasterization → Fragment Shader → Deep Learning Inference → Postprocess → Output

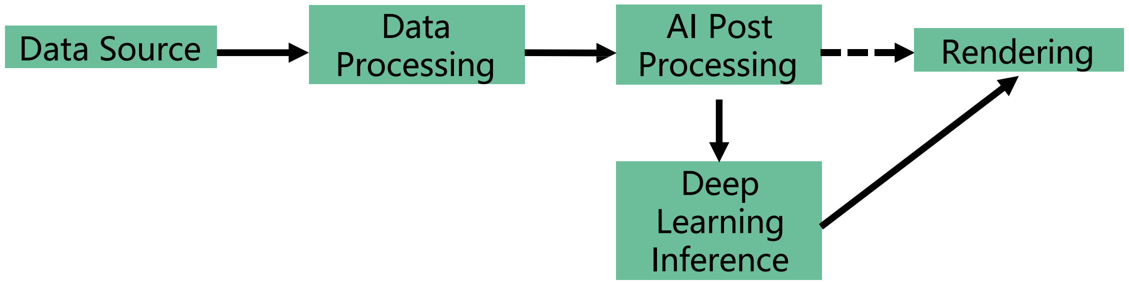
Graphics Applications

Innovations of ShaderNN

- Use **texture-based input/output**, which provides an efficient, zero-copy integration with real-time graphics pipeline or image processing applications, thereby saving expensive data transfers & format conversion between CPU and GPU.



A. Integrate with other inference engines



B. Integrate with ShaderNN

- Leverage the **fragment shader** based on OpenGL backend in the neural network inference operators, which is advantageous when deploying parametrically small neural network modes.
- Built on **native OpenGL ES and Vulkan**, which can be easily integrated with the graphics rendering pipeline to maximize the use of computing resources, suits for rendering, image/video and game AI applications.

- Enable a **hybrid implementation of compute and fragment shaders**, with the ability to select layer-level shaders for performance optimization.

Agenda

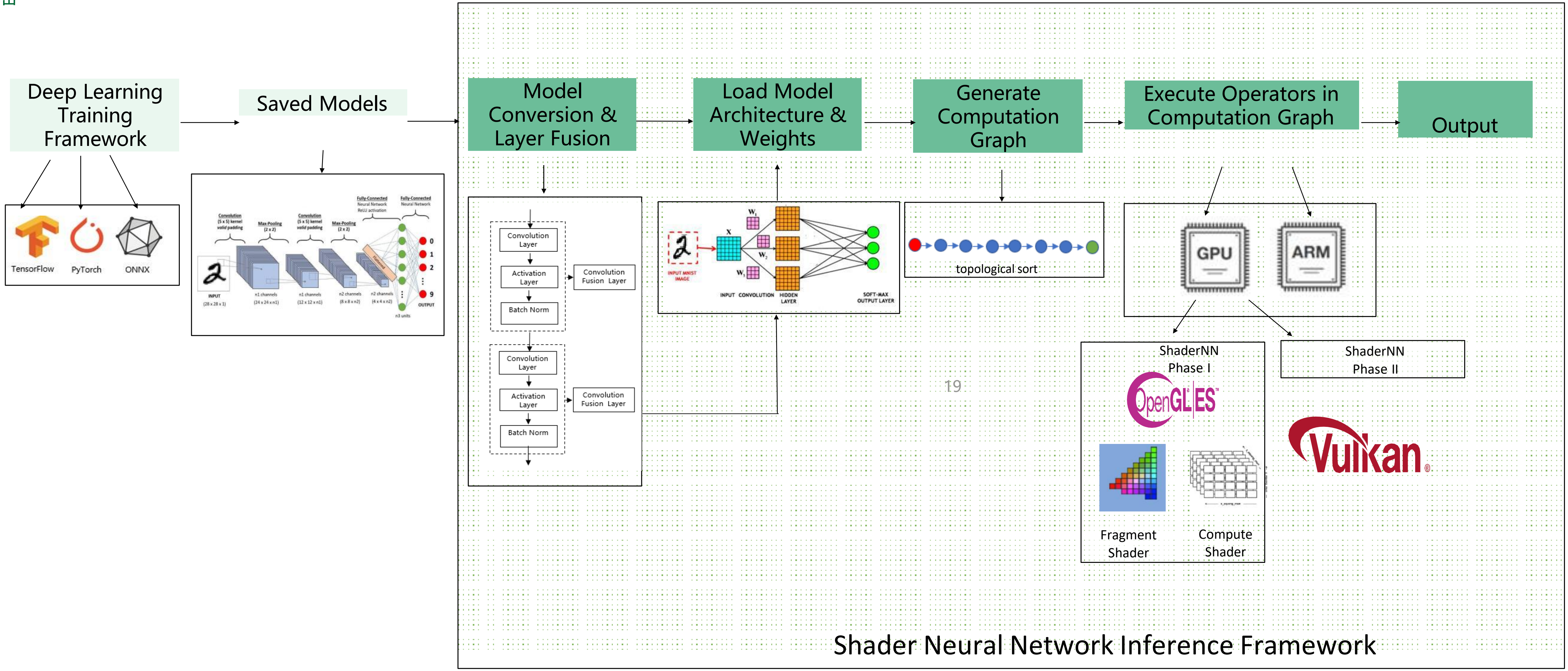
1 Why donate to LF AI & Data

2 Challenges for Mobile Inference

3 What is ShaderNN?

4 ShaderNN Open Source & Roadmap

ShaderNN Workflow



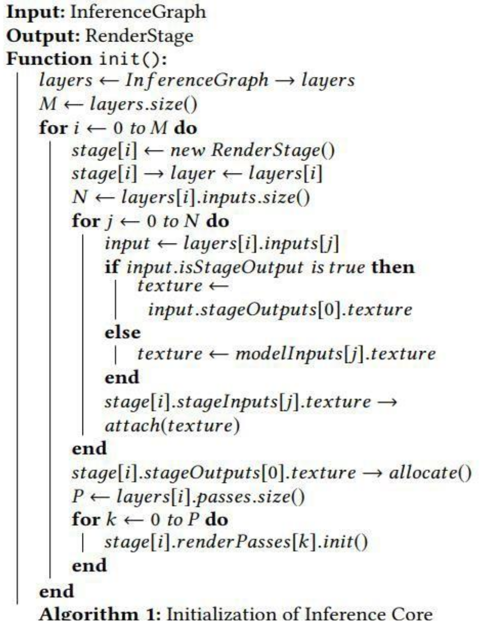
Deep Learning Training Framework → Saved Models → Model Conversion & Layer Fusion → Load Model Architecture & Weights → Generate Computation Graph → Execute Operators in Computation Graph → Output

Shader Neural Network Inference Framework

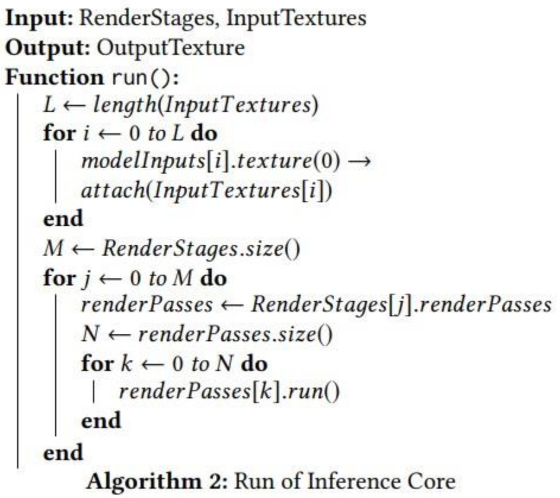
ShaderNN Framework Architecture

<table>
<tr><td rowspan="3">Model Preparation</td><td>Framework</td><td colspan="4">TensorFlow</td><td colspan="5">PyTorch</td><td colspan="5">ONNX</td></tr>
<tr><td>Conversion Tool</td><td colspan="4">TensorFlowConverter</td><td colspan="5">PyTorchConverter</td><td colspan="5">ONNX Converter</td></tr>
<tr><td>Model Optimizations</td><td colspan="2">Model Compressions</td><td colspan="6">Layer Fusion</td><td colspan="4">Grouping Optimization</td><td colspan="2">Operator Optimization</td></tr>
<tr><td rowspan="14">Inference Engine</td><td>Inference Graph</td><td colspan="6">Computation Graph Generation</td><td colspan="8">Topological Sort Schedule</td></tr>
<tr><td>Compile Optimization</td><td colspan="6">Shader Optimization</td><td colspan="8">Equivalent Layers Fusion</td></tr>
<tr><td>Runtime Optimization</td><td>Convolutional Optimization</td><td colspan="4">Texture Reuse</td><td colspan="4">Multi Thread</td><td colspan="4">CPU、GPU Memory Reuse</td><td>C4 Data Layout Cache Vectorization</td></tr>
<tr><td rowspan="11">Supported Operators</td><td colspan="3"></td><td colspan="4">OpenGL Fragment Shader</td><td colspan="3">OpenGL Compute Shader</td><td>CPU</td><td colspan="3">Vulkan Compute Shader</td></tr>
<tr><td colspan="3">Conv2D</td><td colspan="4">X</td><td colspan="3">X</td><td></td><td colspan="3">X</td></tr>
<tr><td colspan="3">Conv2DTranspose</td><td colspan="4">X</td><td colspan="3"></td><td></td><td colspan="3"></td></tr>
<tr><td colspan="3">DepthwiseConv2D</td><td colspan="4">X</td><td colspan="3">X</td><td></td><td colspan="3">X</td></tr>
<tr><td colspan="3">Concatenate</td><td colspan="4">X</td><td colspan="3">X</td><td></td><td colspan="3">X</td></tr>
<tr><td colspan="3">Add</td><td colspan="4">X</td><td colspan="3">X</td><td></td><td colspan="3">X</td></tr>
<tr><td colspan="3">Average Pooling</td><td colspan="4">X</td><td colspan="3">X</td><td></td><td colspan="3">X</td></tr>
<tr><td colspan="3">Max Pooling</td><td colspan="4">X</td><td colspan="3">X</td><td></td><td colspan="3">X</td></tr>
<tr><td colspan="3">Flatten</td><td colspan="4"></td><td colspan="3">X</td><td>X</td><td colspan="3">X</td></tr>
<tr><td colspan="3">Dense</td><td colspan="4"></td><td colspan="3">X</td><td>X</td><td colspan="3">X</td></tr>
<tr><td colspan="3">Upsampling</td><td colspan="4">X</td><td colspan="3">X</td><td></td><td colspan="3">X</td></tr>
</table>

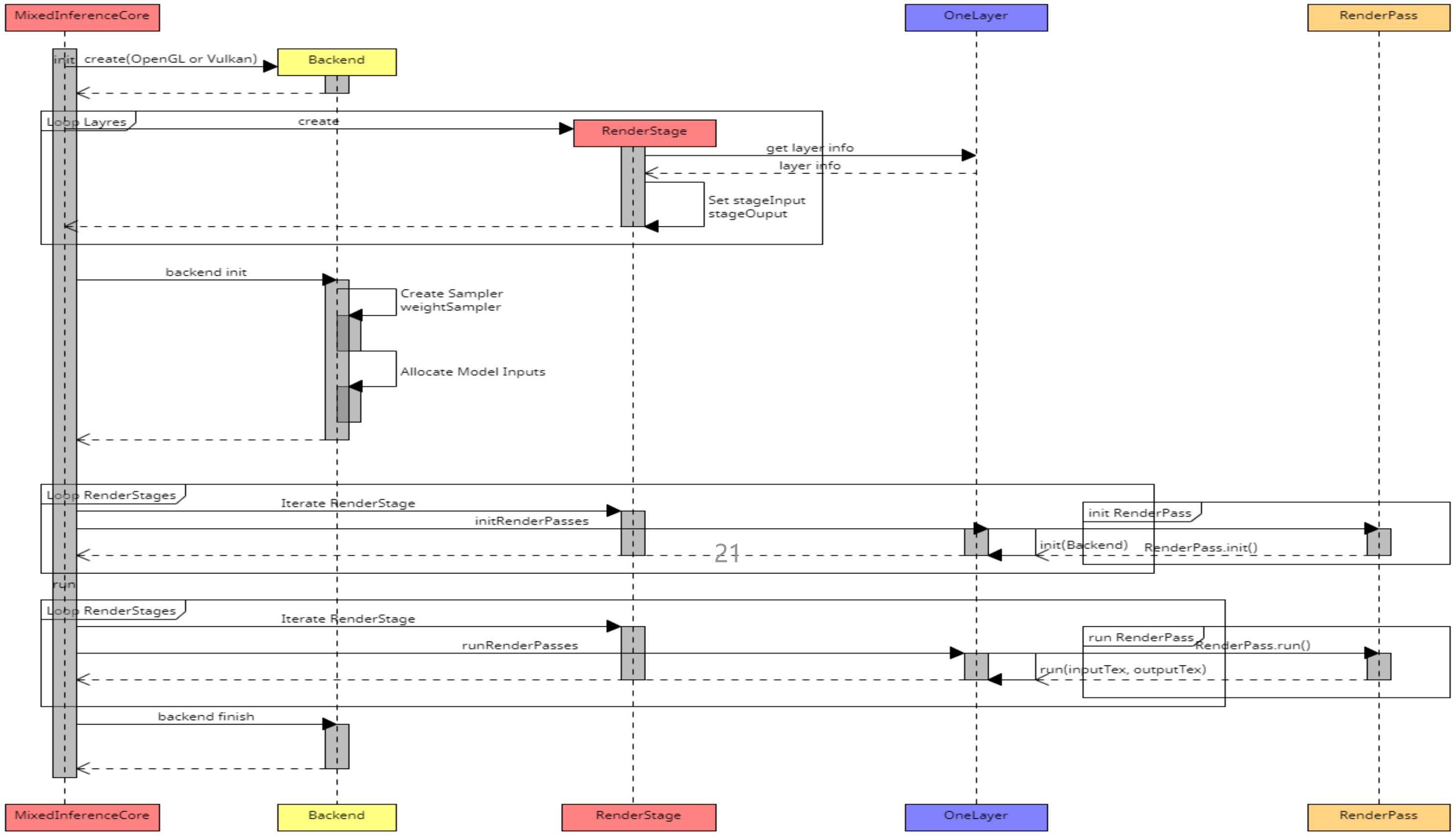
ShaderNN Inference Core Algorithms



```
Input: InferenceGraph
Output: RenderStage
Function init():
    $layers \leftarrow InferenceGraph \rightarrow layers$
    $M \leftarrow layers.size()$
    for $i \leftarrow 0$ to $M$ do
        $stage[i] \leftarrow new\ RenderStage()$
        $stage[i] \rightarrow layer \leftarrow layers[i]$
        $N \leftarrow layers[i].inputs.size()$
        for $j \leftarrow 0$ to $N$ do
            $input \leftarrow layers[i].inputs[j]$
            if $input.isStageOutput$ is $true$ then
                $texture \leftarrow$
                 $input.stageOutputs[0].texture$
            else
                $texture \leftarrow modelInputs[j].texture$
            end
            $stage[i].stageInputs[j].texture \rightarrow$
            $attach(texture)$
        end
        $stage[i].stageOutputs[0].texture \rightarrow allocate()$
        $P \leftarrow layers[i].passes.size()$
        for $k \leftarrow 0$ to $P$ do
            $stage[i].renderPasses[k].init()$
        end
    end
    Algorithm 1: Initialization of Inference Core
```



```
Input: RenderStages, InputTextures
Output: OutputTexture
Function run():
    $L \leftarrow length(InputTextures)$
    for $i \leftarrow 0$ to $L$ do
        $modelInputs[i].texture(0) \rightarrow$
        $attach(InputTextures[i])$
    end
    $M \leftarrow RenderStages.size()$
    for $j \leftarrow 0$ to $M$ do
        $renderPasses \leftarrow RenderStages[j].renderPasses$
        $N \leftarrow renderPasses.size()$
        for $k \leftarrow 0$ to $N$ do
            $renderPasses[k].run()$
        end
    end
        Algorithm 2: Run of Inference Core
```



Key Features of ShaderNN

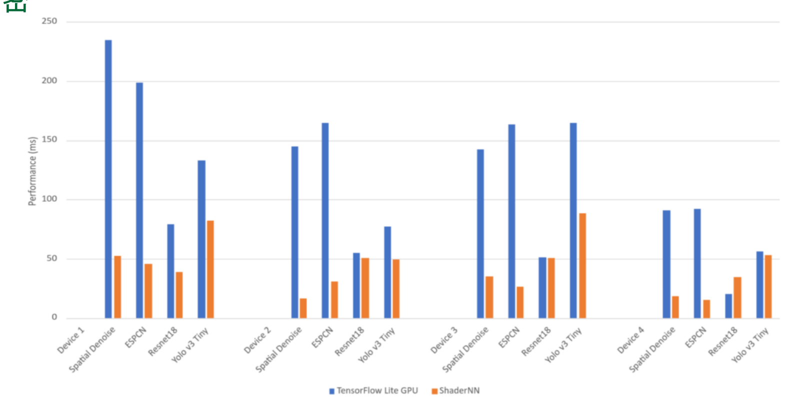
Key Features of ShaderNN

- **High Performance**
  - **Utilize GPU Shader**: Implement core operators using GPU Shader to leverage parallel computing capabilities for optimal performance.
  - **Pre-built Static Computation Graph:** Optimize with constant folding and operator fusion to accelerate forward operation speed.

- **Lightweight & Portability & Extensibility**
  - **No Third-Party Library Dependencies:** Ensure independence from external libraries, reducing overhead and simplifying integration.
  - **Mobile Platform Optimization:** Optimize specifically for mobile platforms, enabling effortless portability, deployment, and upgrades.
  - **Simple Input/Output Interface:** Provide a user-friendly interface compatible with GPU processing for streamlined interactions.

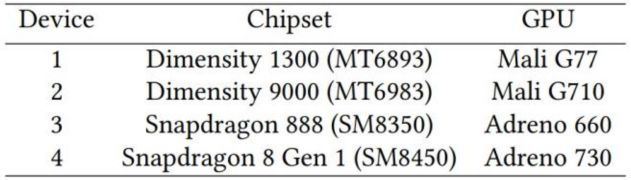
- **Versatility**
  - **Framework & CNN network Compatibility:** Support popular framework formats like TensorFlow, PyTorch, and ONNX. Support common classification, detection, segmentation, and enhancement networks.
  - **User-Defined Operators:** Enable easy implementation of new models by supporting user-defined operators.
  - **Flexible backend configure:** Select the running backend statically or dynamically according to the platform resources during model execution, dynamically adjusting kernel running parameters for minimal energy consumption at runtime.

ShaderNN Performance and Power Consumption Comparison – OpenGL backend with TensorFlow Lite

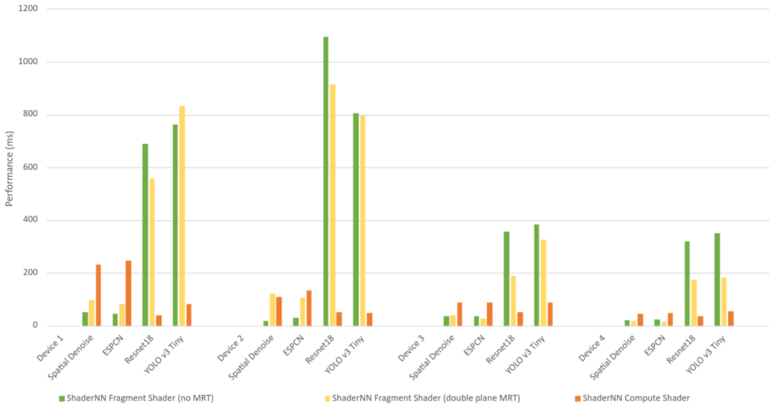


Performance comparison

- On selected target processor chipsets, ShaderNN outperforms TensorFlow Lite on certain tasks, with 75%-90% better performance on spatial denoise and ESPCN, and up to 50% better performance on Resnet18 and YOLO v3 tiny.

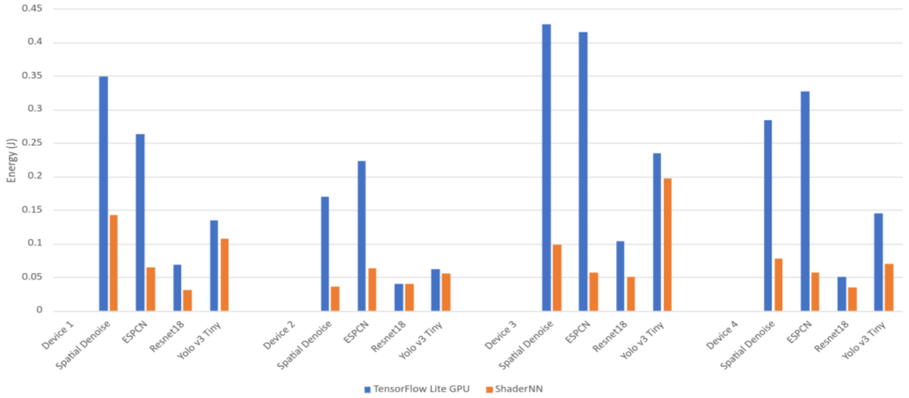


Device	Chipset	GPU
1	Dimensity 1300 (MT6893)	Mali G77
2	Dimensity 9000 (MT6983)	Mali G710
3	Snapdragon 888 (SM8350)	Adreno 660
4	Snapdragon 8 Gen 1 (SM8450)	Adreno 730



Performance comparison over MRT and Fragment/Compute Shader

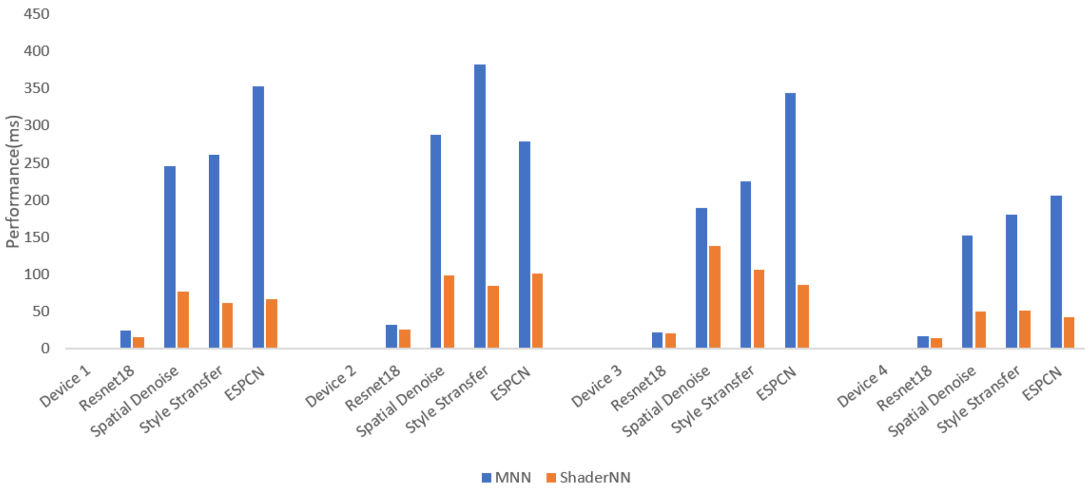
- The fragment shader pipeline offers the option to execute as either no MRT (single render target) or double plane MRT.
- On certain Qualcomm chipsets like Snapdragon SM8350 and SM8450, MRT optimization can provide additional speed up.



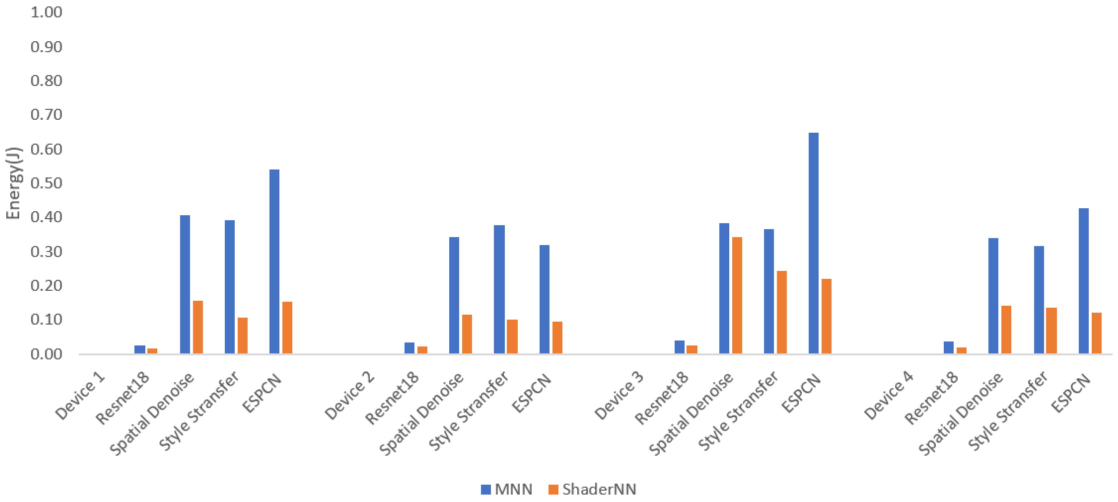
Power consumption comparison

- When inferring Spatial Denoise, ESPCN, Resnet18, and YOLO v3 tiny, ShaderNN can save up to 80%, 70%, 55%, and 51% of energy, respectively.

ShaderNN Performance and Power Consumption Comparison – Vulkan backend with MNN



Performance comparison



Power consumption comparison

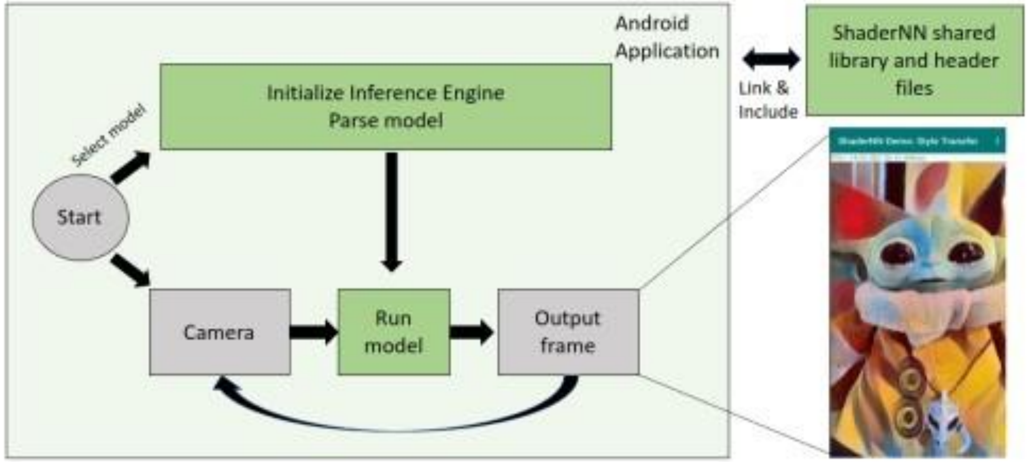
- ShaderNN outperforms MNN on selected target processor chipsets, with 50%-80% better performance on tasks such as spatial denoise and ESPCN, and 6%-60% better performance on tasks such as Resnet18 and Style Transfer.

- When inferring tasks such as Spatial Denoise, ESPCN, Resnet18, and Style Transfer, ShaderNN can save up to 60%, 70%, 45%, and 70% of energy, respectively.

Device	Chipset	GPU
1	Snapdragon 8 Gen 1(SM8450)	Adreno 730
2	Snapdragon 8 Gen 2(SM8550)	Adreno 740
3	Dimensity 9000 (MT6983)	Mali G710
4	Dimensity 9200 (MT6985)	Mali G715

ShaderNN Android Demo App

- A demo app pipeline optimized for throughput over latency, data transfer, and video processing.

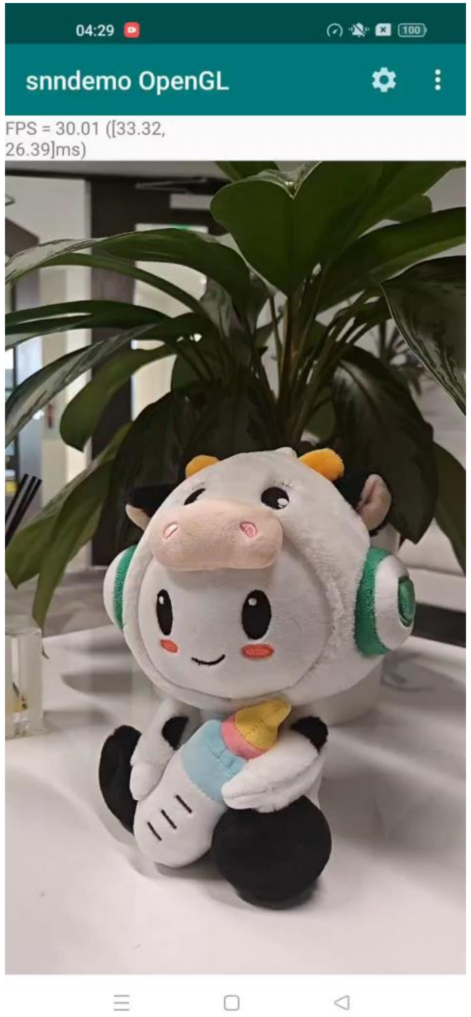


A: Rain Princess Style

B: Udnie Style

C: Candy Style

D: Mosaic Style

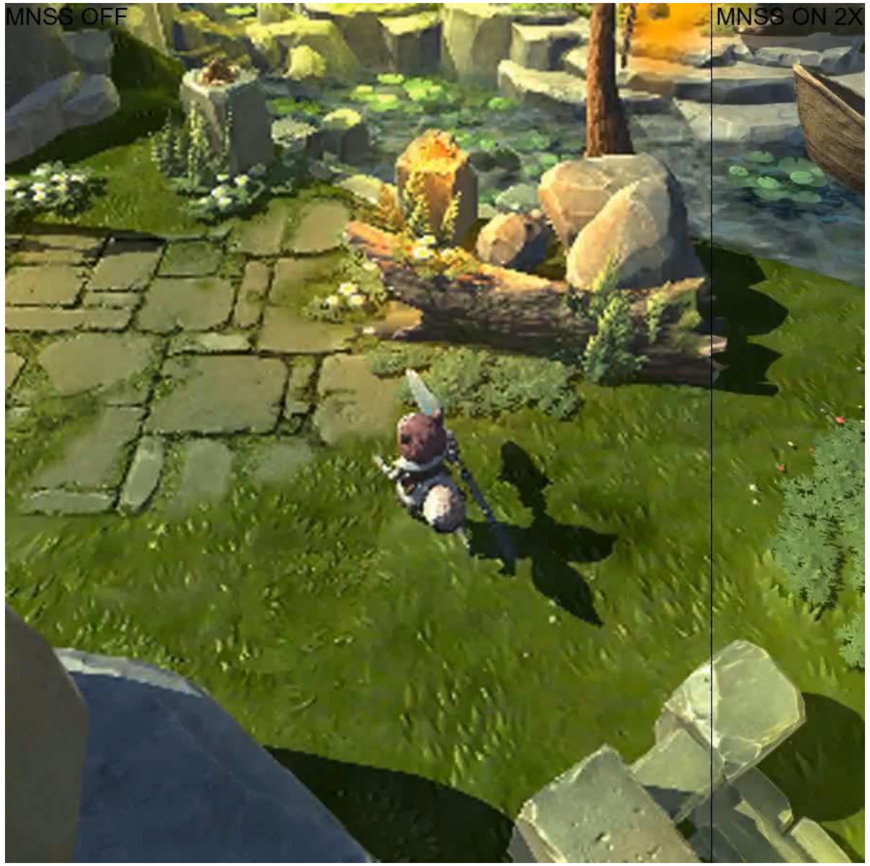


FPS = 30.01 ([33.32, 26.39]ms)

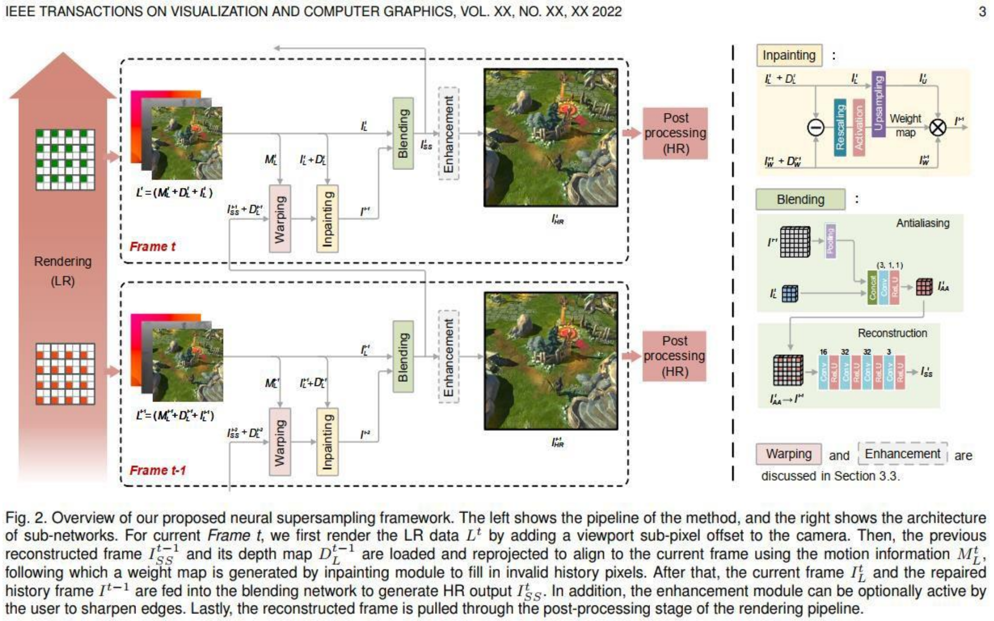
Fast Neural Style Transfer described in Perceptual Losses for Real-Time Style Transfer and Super-Resolution along with Instance Normalization

Cooperation between Academia and Industry

Cooperation between Academia and Industry



MOBA Game 2X Demo



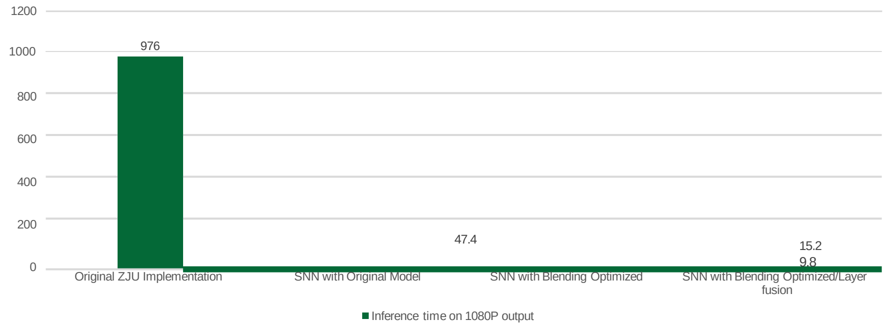
IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL. XX, NO. XX, XX 2022 3

Fig. 2. Overview of our proposed neural supersampling framework. The left shows the pipeline of the method, and the right shows the architecture of sub-networks. For current *Frame t*, we first render the LR data L^t by adding a viewport sub-pixel offset to the camera. Then, the previous reconstructed frame I_{SS}^{t-1} and its depth map D_L^{t-1} are loaded and reprojected to align to the current frame using the motion information M_L^t, following which a weight map is generated by inpainting module to fill in invalid history pixels. After that, the current frame I_L^t and the repaired history frame I^{t-1} are fed into the blending network to generate HR output I_{SS}^t. In addition, the enhancement module can be optionally active by the user to sharpen edges. Lastly, the reconstructed frame is pulled through the post-processing stage of the rendering pipeline.

MNSS: Neural Supersampling Framework for Real-Time Rendering on Mobile Devices

by Zhejiang University and OPPO

MNSS v2 on Qualcomm 8 Gen 2 (ms)



文件密级：秘密

Agenda

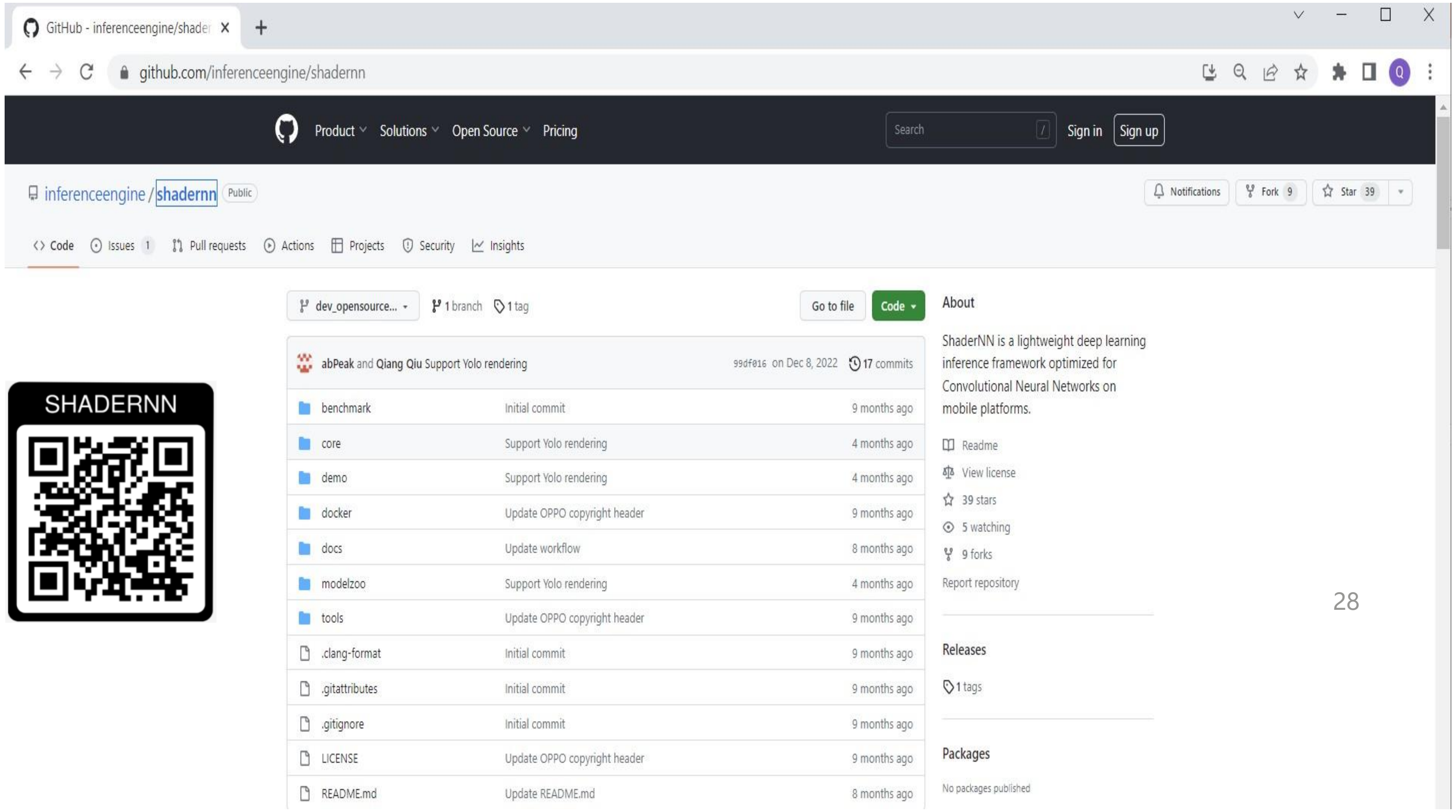
1 Why donate to LF AI & Data

2 Challenges for Mobile Inference

3 What is ShaderNN?

4 ShaderNN Open Source & Roadmap

ShaderNN Open Source



abPeak and Qiang Qiu Support Yolo rendering 99df016 on Dec 8, 2022 17 commits

benchmark	Initial commit	9 months ago
core	Support Yolo rendering	4 months ago
demo	Support Yolo rendering	4 months ago
docker	Update OPPO copyright header	9 months ago
docs	Update workflow	8 months ago
modelzoo	Support Yolo rendering	4 months ago
tools	Update OPPO copyright header	9 months ago
.clang-format	Initial commit	9 months ago
.gitattributes	Initial commit	9 months ago
.gitignore	Initial commit	9 months ago
LICENSE	Update OPPO copyright header	9 months ago
README.md	Update README.md	8 months ago

About

ShaderNN is a lightweight deep learning inference framework optimized for Convolutional Neural Networks on mobile platforms.

SHADERNN

https://github.com/inferenceengine/shadernn (Apache2.0 License)

- Source Code
  - Standalone inference core that can be easily integrated
- Developer Guide
  - Getting started
  - How to create custom layer
  - How to implement model processor
  - How to load and run model
  - How to validate results
  - How to benchmark
- Tools
  - Tool to covert models from TensorFlow, PyTorch and ONNX
- Demo App
  - Provide Android demo app to show how to integrate ShaderNN
- Model Zoo
  - Provide common CNN models

ShaderNN Roadmap

ShaderNN Roadmap

2021.10 – 2022.6 ShaderNN Phase I	2022.7-2023.5 ShaderNN Phase II	2023.6-2023.12 ShaderNN Phase III
1. Support OpenGL Fragment Shader backend 2. Support OpenGL Compute Shader backend 3. Open source ShaderNN 1.0 with Apache 2.0 License 4. Demonstrate ShaderNN features at SIGGRAPH 2022	1. Support Vulkan Compute Shader backend 2. Support multiple inputs 3. Open source ShaderNN 2.0 preview release 4. Integrate into OPPO inference platform framework	1. Join LFAI & DATA Sandbox program 2. Demonstrate ShaderNN new features at SIGGRAPH 2023 3. Add new operator support 4. Add new model conversion support 5. Optimize convolution and matrix multiplication 6. Optimize scheduling that automatically selects backend 7. Engage more ShaderNN users

Future Work

- Companies that may be invited as maintainers for the open-source community
  - MediaTek
  - Qualcomm
  - Universities, such as Zhejiang University
- Key technical points for co-construction.
  - New operator and model support
  - ARM optimization
  - OpenGL and Vulkan backend optimization
  - AIGC applications
- Key product demo & implementations
  - Deep learning Super Sampling for mobile game
- Potential target users
  - Mobile GPU providers
  - Android AI app developers
  - University researchers

Possible Collaboration with LF AI & Data Projects

Possible Collaboration with LF AI & Data Projects

- Integrate data lineage with ONNX and OpenBytes.
- Potentially be integrated as a middleware plugin for end-side graphics-accelerated computations by Adlik and DeepRec.
- As a friendly tech community to share optimization points for graphics acceleration technology with BeyondML and Acumos AI.

We are requesting your support to host ShaderNN in LF AI & Data as a Sanbox Project

Thank you

oppo

Approval of ShaderNN as a Sandbox project

Proposed Resolution:

- ShaderNN as a Sandbox project of the LF AI & Data Foundation is hereby approved.

Upcoming TAC Meetings

LF AI & DATA

Upcoming TAC Meetings

- July 29 – Docarry proposal to move from Sandbox to Incubation, Tentative Project review
- August 10 - LF Edge Presentation

Please note we are always open to special topics as well.

If you have a topic idea or agenda item, please send agenda topic requests to tac-general@lists.lfaidata.foundation

Open Discussion

LF AI & DATA

TAC Meeting Details

- To subscribe to the TAC Group Calendar, visit the wiki: https://wiki.lfaidata.foundation/x/cQB2 ______
- Join from PC, Mac, Linux, iOS or Android: https://zoom.us/j/430697670
- Or iPhone one-tap:
  - US: +16465588656,,430697670# or +16699006833,,430697670#
- Or Telephone:
  - Dial(for higher quality, dial a number based on your current location):
  - US: +1 646 558 8656 or +1 669 900 6833 or +1 855 880 1246 (Toll Free) or +1 877 369 0926 (Toll Free)
- Meeting ID: 430 697 670
- International numbers available: https://zoom.us/u/achYtcw7uN

Legal Notice

- The Linux Foundation, The Linux Foundation logos, and other marks that may be used herein are owned by The Linux Foundation or its affiliated entities, and are subject to The Linux Foundation's Trademark Usage Policy at https://www.linuxfoundation.org/trademark-usage, as may be modified from time to time.
- Linux is a registered trademark of Linus Torvalds. Please see the Linux Mark Institute's trademark usage page at https://lmi.linuxfoundation.org for details regarding use of this trademark.
- Some marks that may be used herein are owned by projects operating as separately incorporated entities managed by The Linux Foundation, and have their own trademarks, policies and usage guidelines.
- TWITTER, TWEET, RETWEET and the Twitter logo are trademarks of Twitter, Inc. or its affiliates.
- Facebook and the "f" logo are trademarks of Facebook or its affiliates.
- LinkedIn, the LinkedIn logo, the IN logo and InMail are registered trademarks or trademarks of LinkedIn Corporation and its affiliates in the United States and/or other countries.
- YouTube and the YouTube icon are trademarks of YouTube or its affiliates.
- All other trademarks are the property of their respective owners. Use of such marks herein does not represent affiliation with or authorization, sponsorship or approval by such owners unless otherwise expressly specified.
- The Linux Foundation is subject to other policies, including without limitation its Privacy Policy at https://www.linuxfoundation.org/privacy and its Antitrust Policy at https://www.linuxfoundation.org/antitrust-policy. each as may be modified from time to time. More information about The Linux Foundation's policies is available at https://www.linuxfoundation.org.
- Please email legal@linuxfoundation.org with any questions about The Linux Foundation's policies or the notices set forth on this slide.

LF AI & DATA

13JUL2023