

PyCon China 2024

For Good . For fun.
2024/11/23 中国 上海



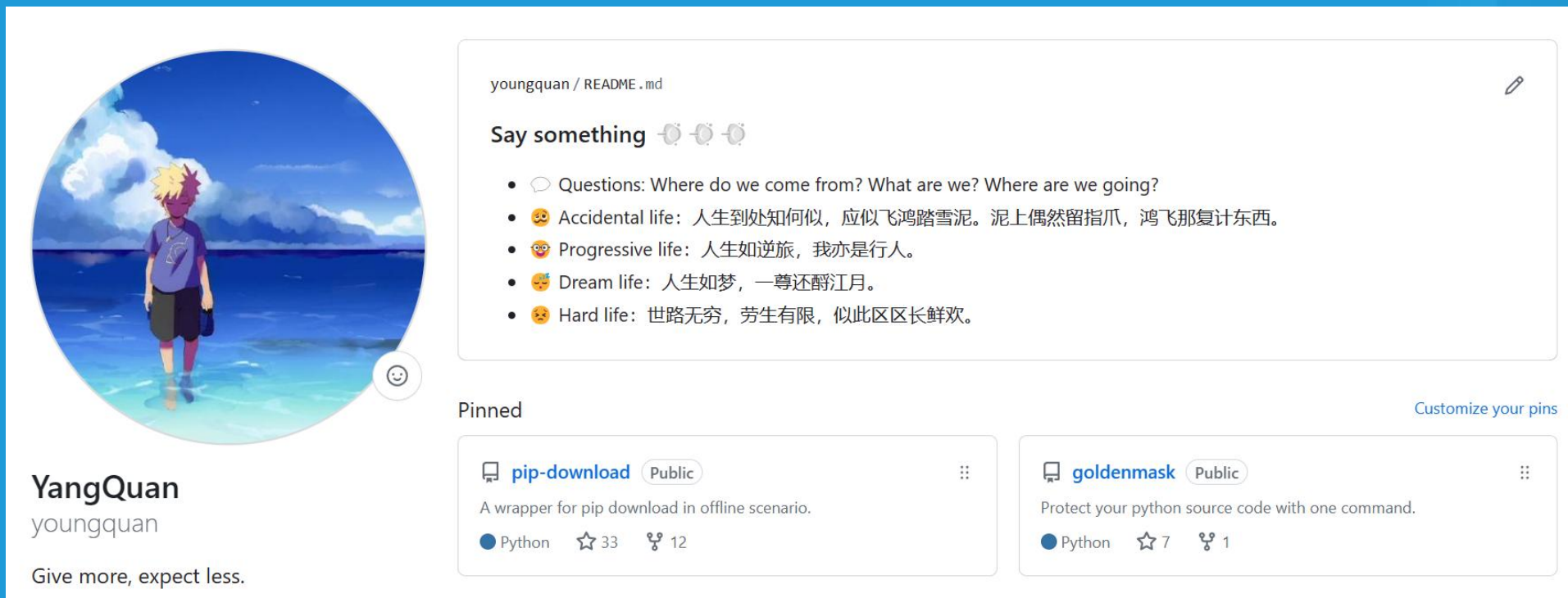
Pytorch 模型优化部署背后的技术探究

杨权



个人介绍

- 过去：兰州大学、西安交通大学、中国电科
- 现在：哈尔滨工业大学重庆研究院 机器视觉工程师
- 未来：XXX CTO 🤖



A screenshot of a GitHub profile for user YangQuan (youngquan). The profile includes a circular avatar of a character standing on a beach, a bio, a pinned README file, and two pinned repositories.

Profile:

- Avatar: A circular image of a character with spiky yellow hair and a purple shirt standing on a beach.
- Name: YangQuan
- Username: youngquan
- Bio: Give more, expect less.

youngquan / README.md

Say something

- Questions: Where do we come from? What are we? Where are we going?
- Accidental life: 人生到处知何似，应似飞鸿踏雪泥。泥上偶然留指爪，鸿飞那复计东西。
- Progressive life: 人生如逆旅，我亦是行人。
- Dream life: 人生如梦，一尊还酹江月。
- Hard life: 世路无穷，劳生有限，似此区区长鲜欢。

Pinned

pip-download Public

A wrapper for pip download in offline scenario.

Python 33 12

goldenmask Public

Protect your python source code with one command.

Python 7 1



技术背景

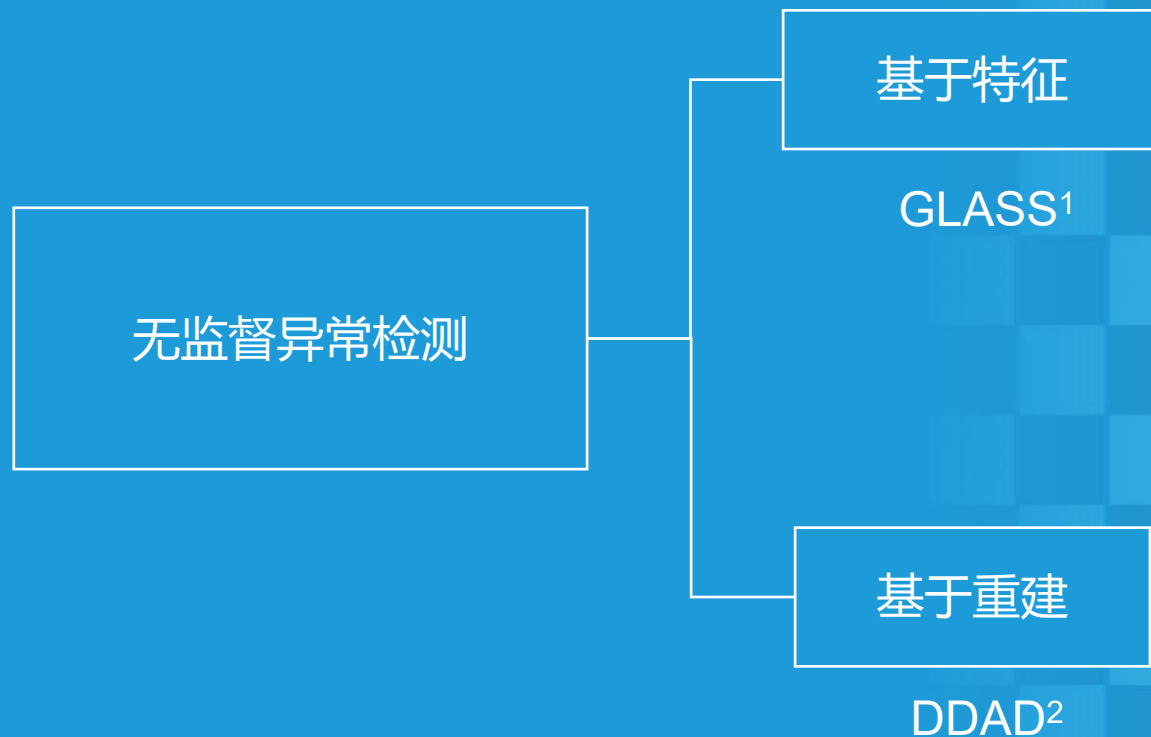


技术分析



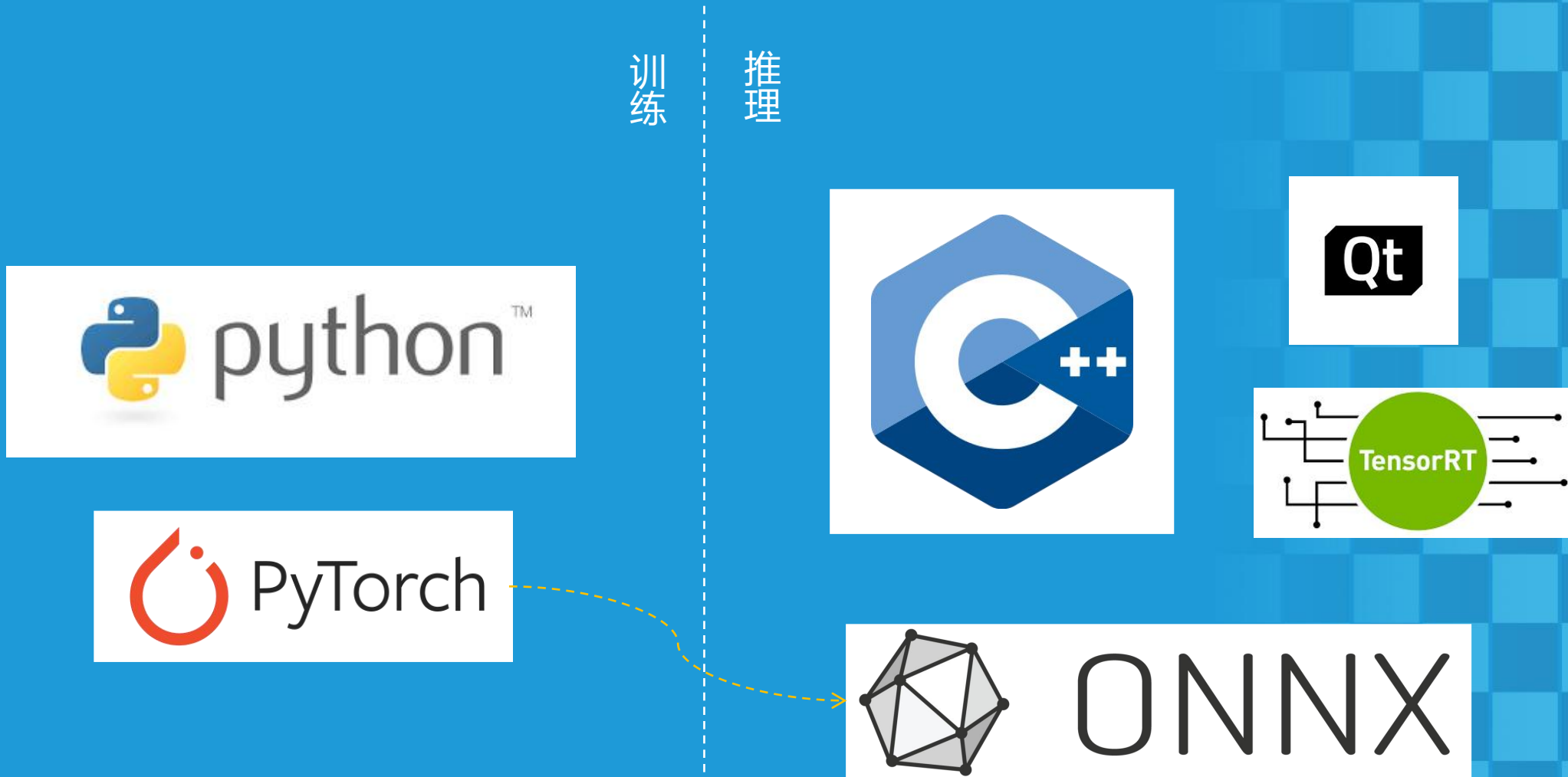
技术应用

工业图像异常检测



1. Chen, Qiyu, et al. "A unified anomaly synthesis strategy with gradient ascent for industrial anomaly detection and localization." arXiv preprint arXiv:2407.09359 (2024).
2. Mousakhan, Arian, Thomas Brox, and Jawad Tayyub. "Anomaly detection with conditioned denoising diffusion models." arXiv preprint arXiv:2305.15956 (2023).

核心技术栈





技术背景

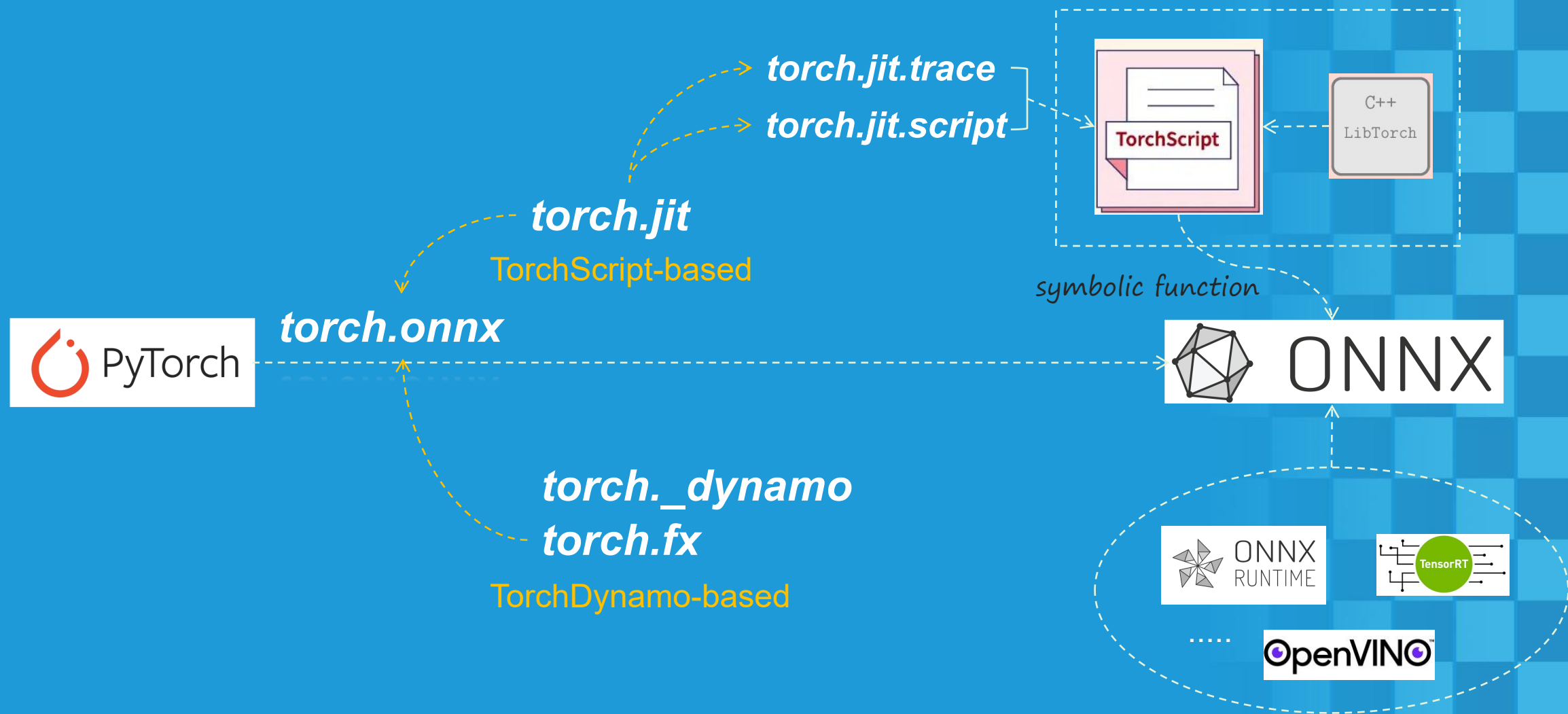


技术分析

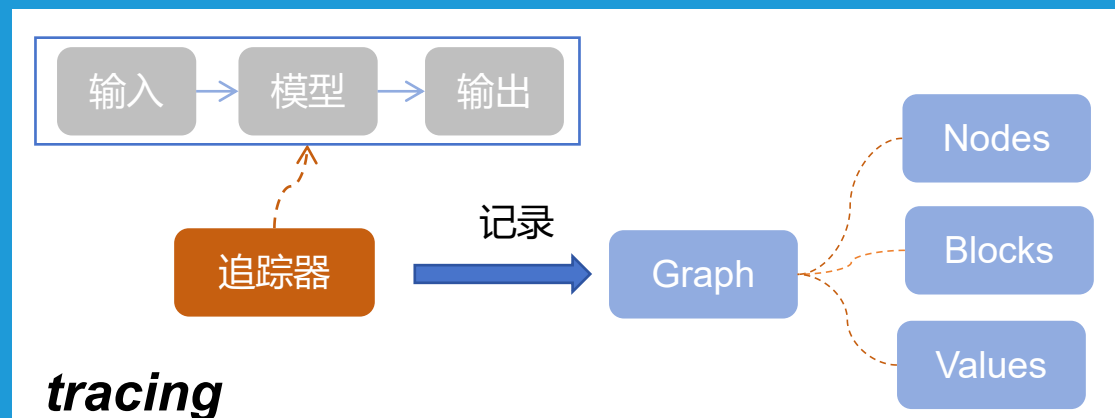


技术应用

Pytorch模型部署技术路线

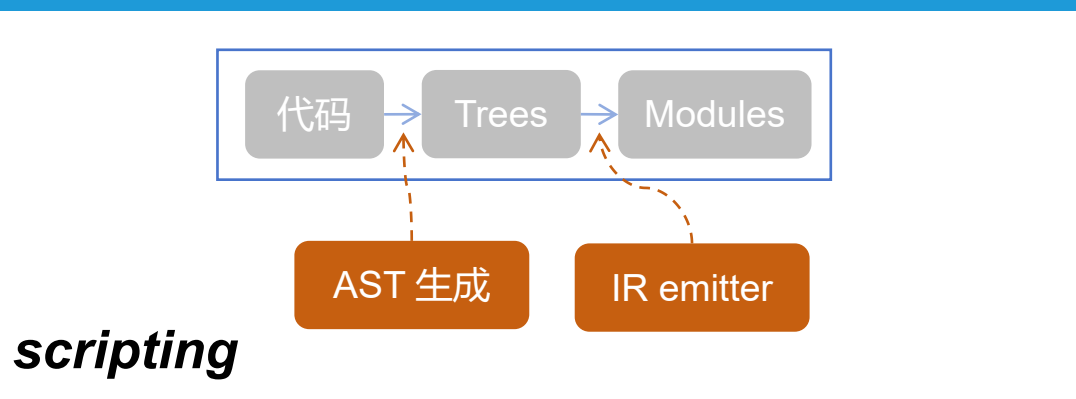


torch.jit



Eager Mode

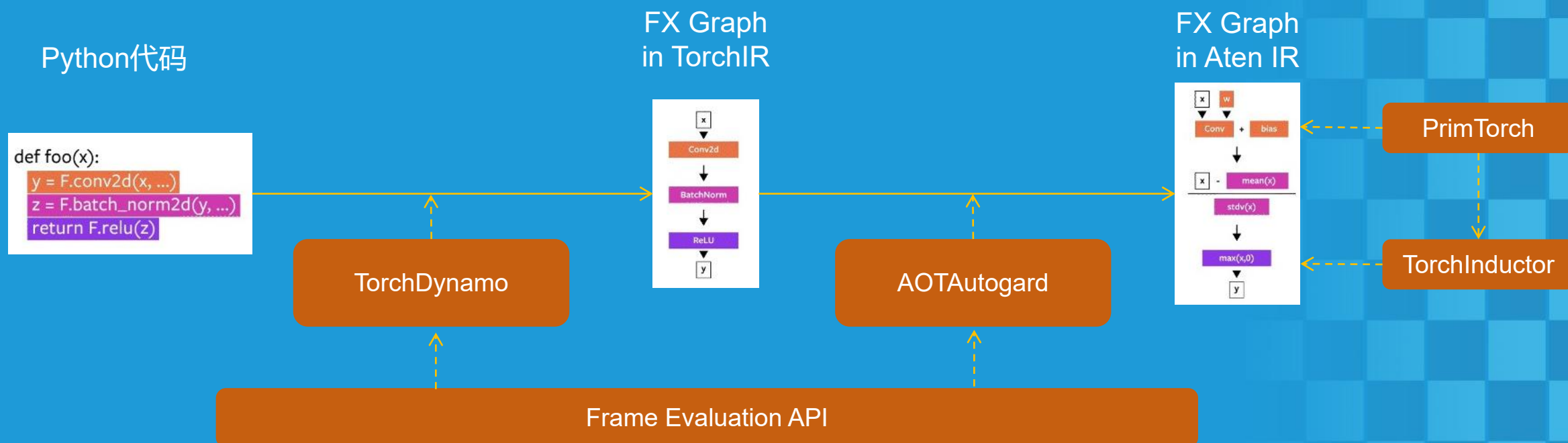
- 动态图



Graph Mode

- 静态图

torch._dynamo + torch.fx





技术背景



技术分析



技术应用

实例一：DDAD

```
torch.onnx.export(  
    ddad,  
    dummy_input,  
    "ddad.onnx",  
    verbose=True,  
    input_names=["img"],  
    output_names=["map"],  
)
```

实例一：DDAD

```
torch.onnx.errors.SymbolicValueError: Unsupported: ONNX export of
convolution for kernel of unknown shape. [Caused by the value
'43357 defined in (%43357 : Float(*, *, *, *, strides=[73728,
73728, 288, 1], requires_grad=0, device=cuda:0) =
onnx::Reshape[allowzero=0] (%43332, %43356), scope: ddad.DDAD_ONNX::
# D:\yangquan\projects\DDAD\.venv\lib\site-
packages\kornia\filters\filter.py:126:0
```

实例一：DDAD

```
input = input.view(-1, tmp_kernel.size(0), input.size(-2), input.size(-1))
```

```
input = input.view(-1, tmp_kernel.size(0), input.size(-2), input.size(-1))
```

When indexing into a tensor for reading, the following patterns are not supported:

```
# Tensor indices that includes negative values.
```

```
data[torch.tensor([[1, 2], [2, -3]]), torch.tensor([-2, 3])]
```


```
# Workarounds: use positive index values.
```

实例一：DDAD

```
def heat_map(output, target, FE, config):  
    sigma = 4  
  
    kernel_size = 2 * int(4 * sigma + 0.5) + 1  
  
    anomaly_map = 0  
  
    output = output.to(config.model.device)  
    target = target.to(config.model.device)  
  
    i_d = pixel_distance(output, target)  
  
    f_d = feature_distance((output), (target), FE, config)  
    f_d = torch.Tensor(f_d).to(config.model.device)  
  
    anomaly_map += f_d + config.model.v * (torch.max(f_d) / torch.max(i_d)) * i_d  
  
    anomaly_map = gaussian_blur2d(  
        anomaly_map, kernel_size=(kernel_size, kernel_size), sigma=(sigma, sigma)  
    )
```

实例一：DDAD

```
input = input.view(-1, tmp_kernel.size(0),  
input.size(-2), input.size(-1))
```

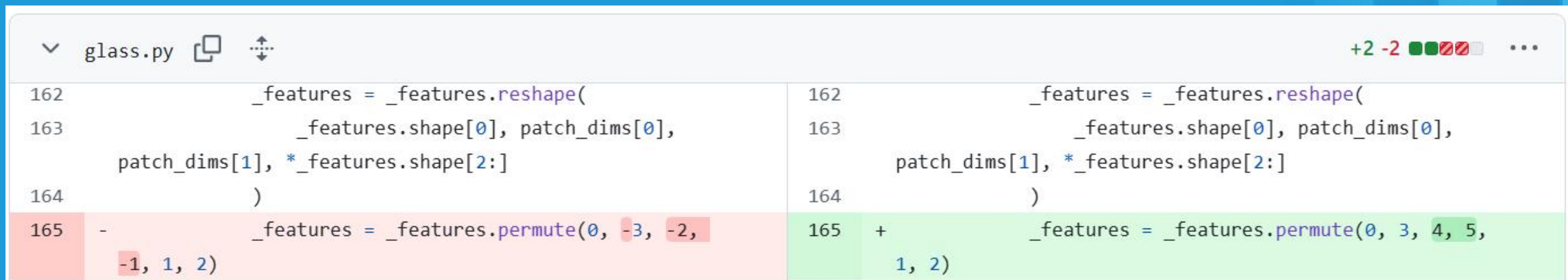


```
def forward(self, input_tensor: torch.Tensor) -> torch.Tensor:  
    batch, channel, height, width = input_tensor.size()  
  
    .....  
  
    output = F.conv2d(  
        input_tensor, self.kernel, groups=self.channels, padding=0, stride=1  
    )  
  
    if self.padding == "same":  
        out = output.view(batch, channel, height, width)  
    else:  
        out = output.view(  
            batch, channel, height - self.height + 1, width - self.width + 1  
        )  
  
    return out
```

[*https://github.com/openvinotoolkit/anomalib](https://github.com/openvinotoolkit/anomalib)

实例二：GLASS

```
torch.onnx.errors.SymbolicValueError: Unsupported: ONNX export of operator adaptive_avg_pool1d, input size not accessible. Please feel free to request support or submit a pull request on PyTorch GitHub:  
https://github.com/pytorch/pytorch/issues [Caused by the value '770' defined in (%770 : Float(*, *, *, strides=[1, 11943936, 1296], requires_grad=0, device=cuda:0) = onnx::Reshape[allowzero=0](%features.3, %769), scope: glass.GLASS::/common.Preprocessing::preprocessing/common.MeanMapper::preprocessing_modules.1 # D:\yangquan\projects\GLASS\common.py:36:0
```



```
glass.py  +2 -2
```

162		<code>_features = _features.reshape(</code>	162		<code>_features = _features.reshape(</code>
163		<code> _features.shape[0], patch_dims[0],</code>	163		<code> _features.shape[0], patch_dims[0],</code>
		<code> patch_dims[1], *_features.shape[2:]</code>			<code> patch_dims[1], *_features.shape[2:]</code>
164		<code>)</code>	164		<code>)</code>
165	-	<code>_features = _features.permute(0, -3, -2,</code>	165	+	<code>_features = _features.permute(0, 3, 4, 5,</code>
		<code>-1, 1, 2)</code>			<code>1, 2)</code>

TorchDynamo-based



youngquan commented yesterday

Contributor ...

Describe the bug

1. Spaces in the Windows path:

I'm using Python installed in `C:\Program Files\Python310`. When running the compilation command, it fails because the path `Program Files` contains a space. The `cl.exe` compiler splits the path into two separate arguments, causing the build to fail.

2. Virtual environment missing required library path:

I created a virtual environment using `python -m venv venv`. However, the build command doesn't include the necessary path to `python310.lib`, located in `C:\Program Files\Python310\libs`. This results in a failure to link against Python during the compilation process.



youngquan commented yesterday • edited

Contributor ...

Fix the issue with `torch.compile` on Windows due to spaces in the path and virtual environment setup.

1. Wrap include paths in double quotes:

To handle paths with spaces, enclose the header file paths in double quotes. This ensures the compiler treats the entire path as a single argument.

2. Adapt library linkage to the virtual environment type:

Dynamically determine the correct library path based on the type of virtual environment being used, ensuring the linker can find and use the necessary libraries like `python310.lib`.

Fixes [#141026](#)

技术无关

- 配置环境好烦人..... 😞
- 模型应用好麻烦..... 😞

- 给我一个可运行该算法的环境..... 😊
- 帮我生成一个API接口..... 😊
- 把模型转成ONNX并提供运行代码..... 😊

生成式AI时代

专注算法

- 新的框架
- 新的语言
- 新的平台
-

谢谢！

