
intel
OpenVINO™
工具套件
基础培训-杨亦诚
安装
使用
示例
visit openvino.ai
1 oneAPI
Powered by oneAPI

Nezha Dev.Kit

哪吒开发套件

基于最新Intel® N97 处理器(原代号Alder Lake-N)

- Intel® N97 处理器
- 板载 LPDDR5 内存，最大8GB
- 板载 eMMC 存储，最大64GB
- 1 Gigabit LAN x 1
- HDMI 1.4b x 1
- USB 3.2 Gen 2 x 3，USB 2.0 x 2
- 40-pin GPIO x 1
- 12V DC-in, 5A
- TPM 2.0

强大的异构计算能力 | 1.8寸信用卡大小 | 工业设计可量产 | 丰富的拓展接口

新世代ATOM®处理器的强大能力

全新的Intel® N97 (原代号Alder Lake-N) 可提供4核最大3.60GHz的主频，比上代单核性能提升至1.4倍。采用英特尔7制程工艺，功耗仅为12W TDP。

Intel® UHD Graphics Gen12

集成的12代图形适配器基于 Xe 架构，可提供最多 24 个运行频率为 1200 MHz 的 EU。搭载HDMI 1.4b显示接口，可支持4K UHD (3840×2160) at 30Hz高清显示。

强大的AI性能，加速AIGC应用

AI性能较上代提升至3.5倍。通过OpenVINO®指令集的加持，可用于加速AI推理以及AIGC的应用。

电源需求	
电源	12V DC-in, 5A
电源规格	AT (default) /ATX
典型功耗	30~36W
物理特性	
尺寸	3.34" x 2.20" (85mm x 56mm)
净重	0.33 lb. (0.15Kg)
毛重	0.44 lb. (0.20Kg)
环境	
工作温度	32°F ~ 140°F (0°C ~ 60°C) / 0.5 airflow
工作湿度	0% ~ 90% relative humidity, non-condensing
MTBF	685,218 Hours
认证	CE/FCC Class A, RoHS Compliant, REACH

系统	
处理器	Intel® Processor N97 (formerly Alder Lake-N)
图形	Intel® UHD Graphics
内存	Dual-Channel LPDDR5, 4GB/8GB
存储	32GB/64GB eMMC
I/O	HDMI 1.4b/USB 3.2 Gen 2 STACK Connector x 1 (Type-A) 4-pin Front Panel x 1 2-pin Fan Wafer x 1 (12V) 2-pin RTC Battery Wafer x 1
USB	USB 3.2 Gen 2 (Type-A) x 3 10-pin USB 2.0 x 2/UART x 1
拓展	40-pin GPIO x 1
显示接口	HDMI 1.4b x 1
网络	1GbE RJ-45 x 1 (Realtek RTL8111H CG)
安全	Onboard TPM 2.0
RTC	Yes
OS支持	Windows® 10 Enterprise LTSC 2021 Linux: Ubuntu 22.04 LTS/Kernel 5.15, Yocto 4.0

www.aaeon.com
400-770-6868

AAEON 研扬科技

哪吒开发套件

小 Nezha 很能打

技术交流群：QQ群：729389778

官方京东店：研扬AAEON旗舰店

全球首款搭载 intel® N97 处理器的1.8寸开发板
(原代号Alder-Lake N)

www.aaeon.com
400-770-6868

研扬科技成立于1992年，是全球先进工业嵌入式计算平台设计、制造和工业4.0与人工智能边缘计算解决方案的领导厂商，为全球OEM/ODM客户，系统集成商和开发者提供定制化解决方案。

AAEON 研扬科技

AI部署无处不在

使用 OpenVINO™ 工具套件最大限度地利用您的硬件，并利用集成加速功能以最佳成本效益在需要的地方运行 AI 工作负载。

边缘

加速物联网设备应用中的推理，适用于零售、工业制造、医疗保健、智慧城市等领域的各种用例—由全系列英特尔®架构提供支持。

AI PC

借助内置在每个英特尔®酷睿™ Ultra处理器的 AI 加速功能，您可以直接基于 AI PC提供引人入胜的全新体验—增强的协作、生产力和创造力。

云

支持由Intel® Xeon®可扩展处理器提供支持的快速增长的云工作负载，该处理器具有内置加速器，可实现更高的单核性能和无与伦比的AI性能。

visit openvino.ai

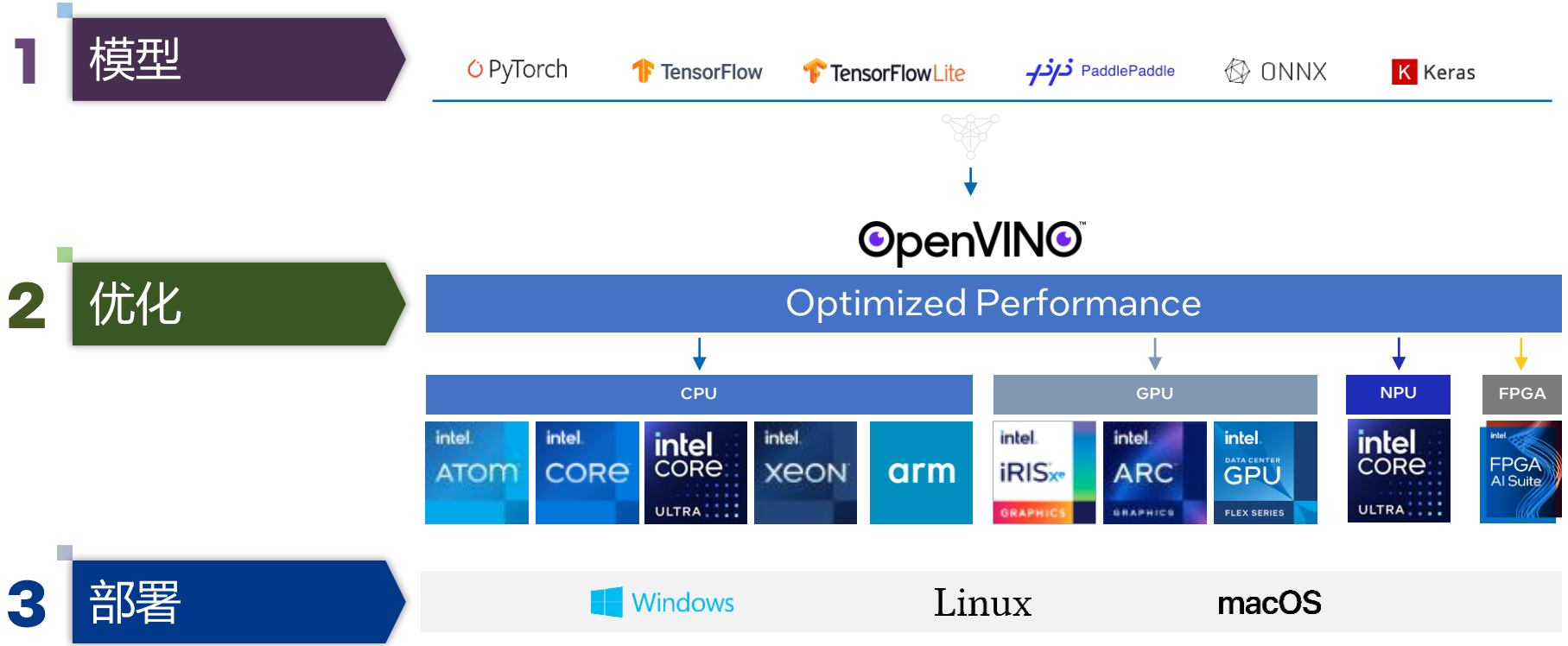
Powered by oneAPI

利用高性能，快速、准确地得到结果

转换和优化模型，并跨硬件进行部署

支持生成式AI、计算机视觉、多模态的数百种模型

使用原生APIs或使用
- Triton Server
- LangChain
- Torch.compile
- Hugging Face Optimum Intel
- 带有OpenVINO后端的ONNX运行时

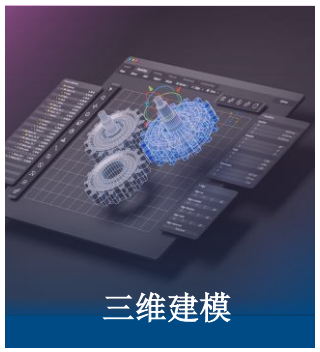


Powered by oneAPI

The productive, smart path to freedom for accelerated computing from the economic and technical burdens of proprietary alternatives.

AI 赋能
使用案例

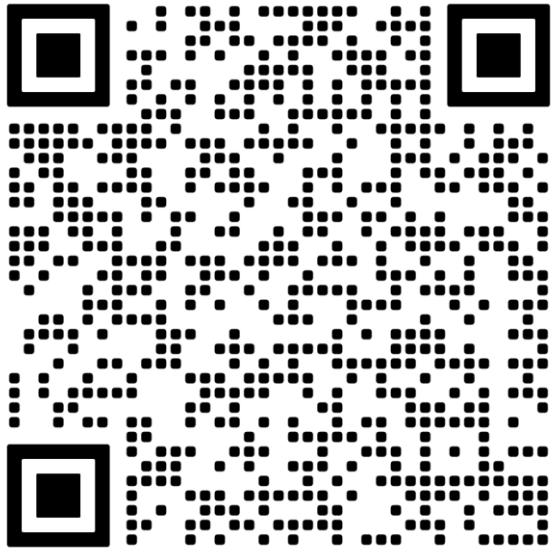
生成式 AI 和大型语言模型



安装

Installation

```
pip install openvino
```



ubuntu

Red Hat

docker

ANACONDA

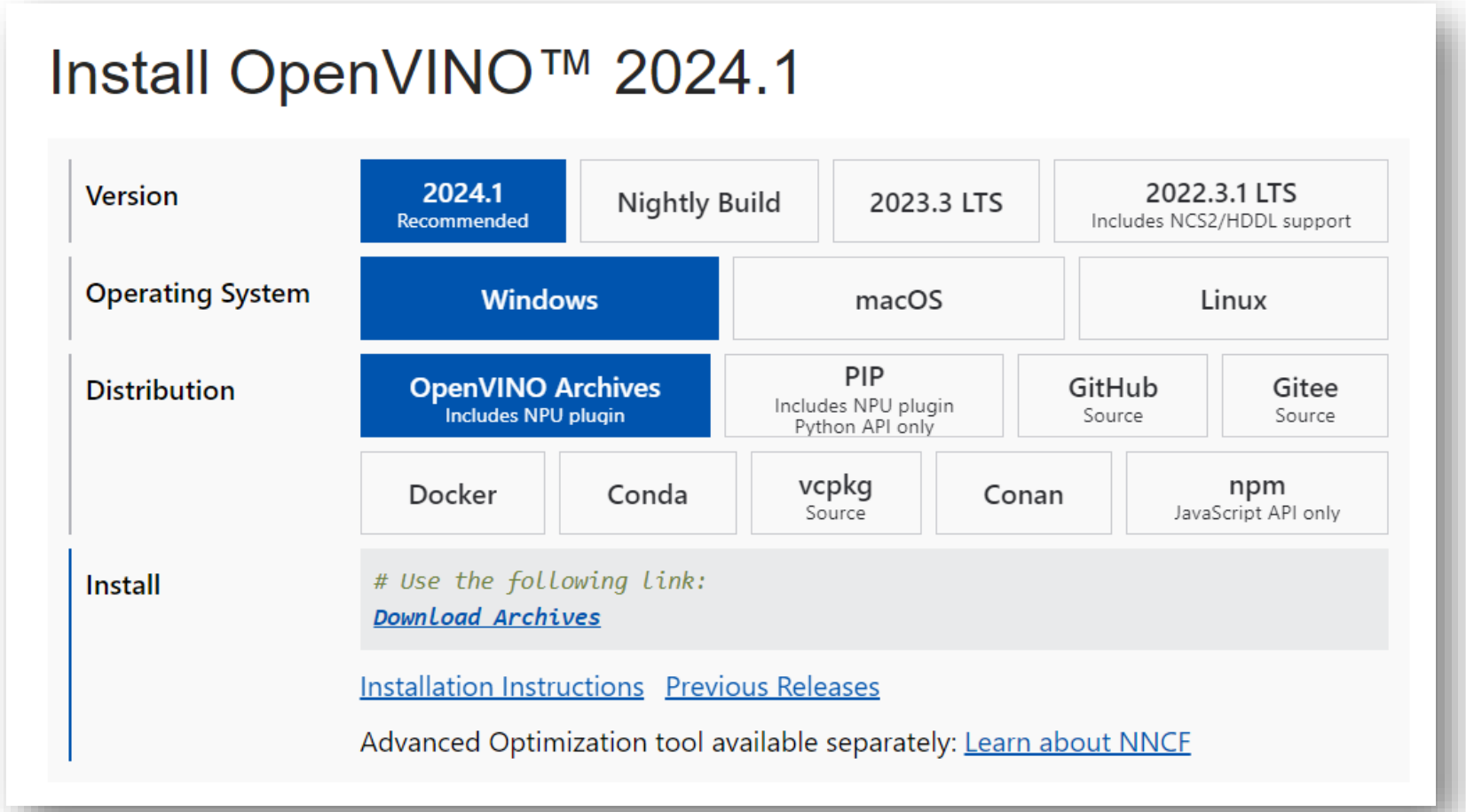
CONDA-FORGE

CONAN
C/C++ Package Manager

npm

www.openvino.ai

Installation



Visual Studio配置示例

“简单三步在Windows上调用低功耗NPU部署AI模型”

使用方式

OpenVINO™ 工具套件开发者之旅

1 | 模型

PyTorch | TensorFlow | TensorFlow Lite | PaddlePaddle | ONNX | Keras

Open Model Zoo
280+ 开源和优化的预训练模型

Intel Optimum
将OpenVINO用作Hugging Face转换器模型中的扩展，并获得模型压缩和性能优势

intel GETi
在短时间内使用更少的数据构建计算机视觉模型

2 | 优化

OpenVINO模型转换器
从支持的框架转换经过训练的模型

IR Data

读取、加载、推理

OpenVINO 格式 (intermediate representation file) (.pb, .tflite, .onnx,)

TensorFlow 和 PyTorch 的直接模型转换
对于选定的模型，可以跳过步骤以更快地进行部署

使用 NNCF 进行模型压缩
神经网络压缩框架提供量化感知训练、模型剪枝和稀疏性以及训练后优化

Jupyter Notebooks
获取有关最新模型的示例代码，以帮助更快地将应用程序投入生产

3 | 部署

OpenVINO™ 模型服务器
通过 gRPC、REST 或 C API 终结点提供模型

OpenVINO™ 运行时
常见的 Python、C 和 C++ API，用于抽象以下每个设备的低级编程

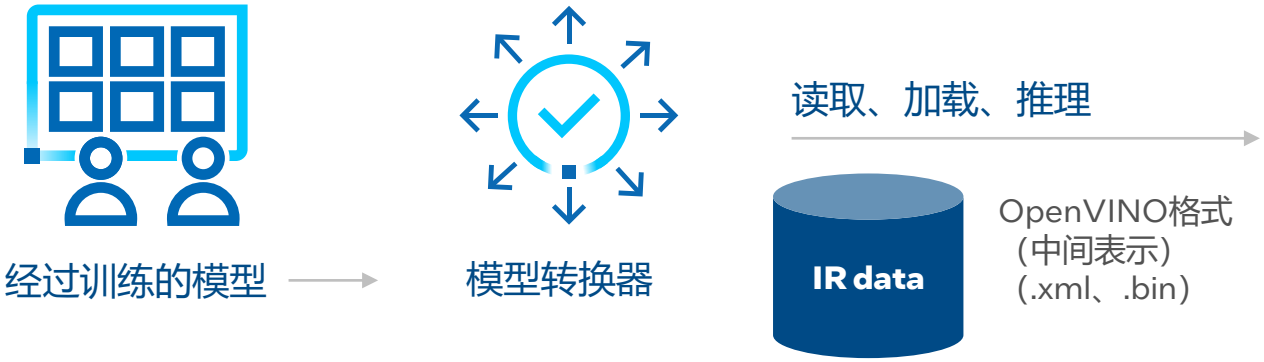
模型转换器

从两个选项中选择:

最佳性能

显式地将模型转换为OpenVINO IR.

模型转换 API 将常用的深度学习操作转换为它们在 OpenVINO 中各自的类似表示形式，并使用来自训练模型的相关权重和偏差进行调整。

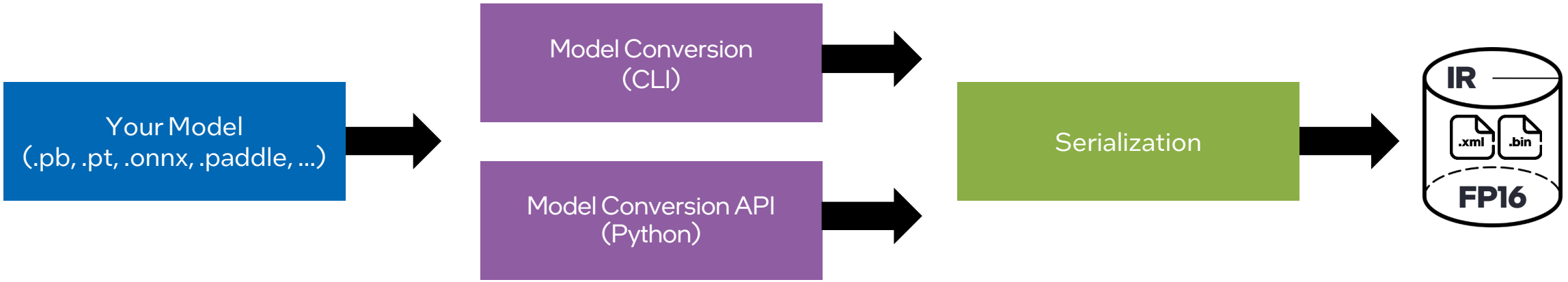


简便方式

跳过 IR 模型转换，直接从源格式运行推理。转换仍将执行，但它将自动发生并“运行在后台”。

此选项虽然方便，但性能和稳定性较低，优化选项也较少。

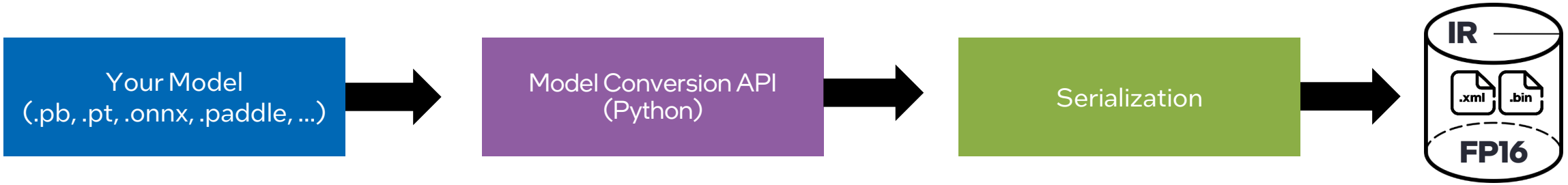
OpenVINO™ 模型转换器



OpenVINO™ 模型转换器

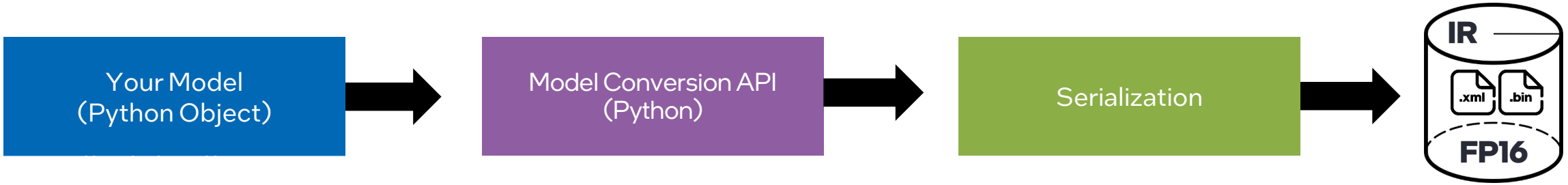

```
ovc "model.onnx" --input "1,3,224,224"
```

OpenVINO™ 模型转换器



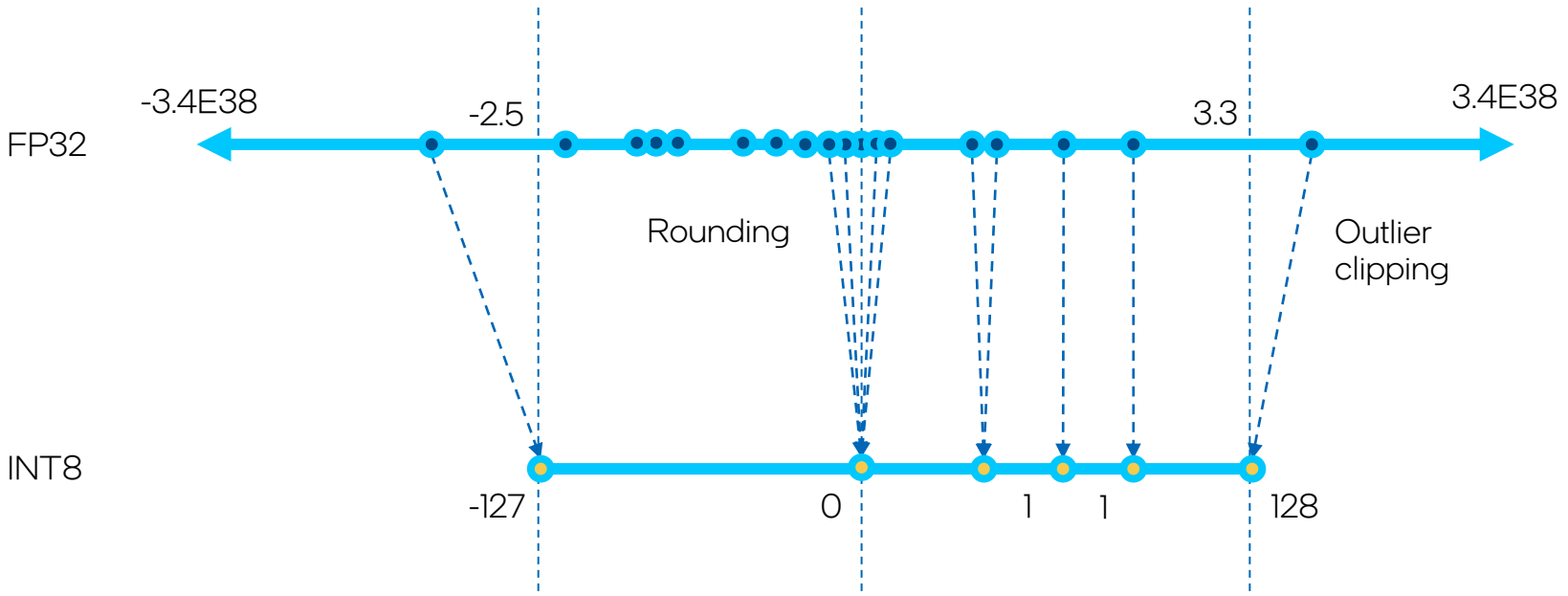

```
ov_model = ov.convert_model("model.onnx",
                            input={"input1": [1,3,224,224]})

ov.save_model(ov_model, "converted_model.xml")
```

OpenVINO™ 模型转换器



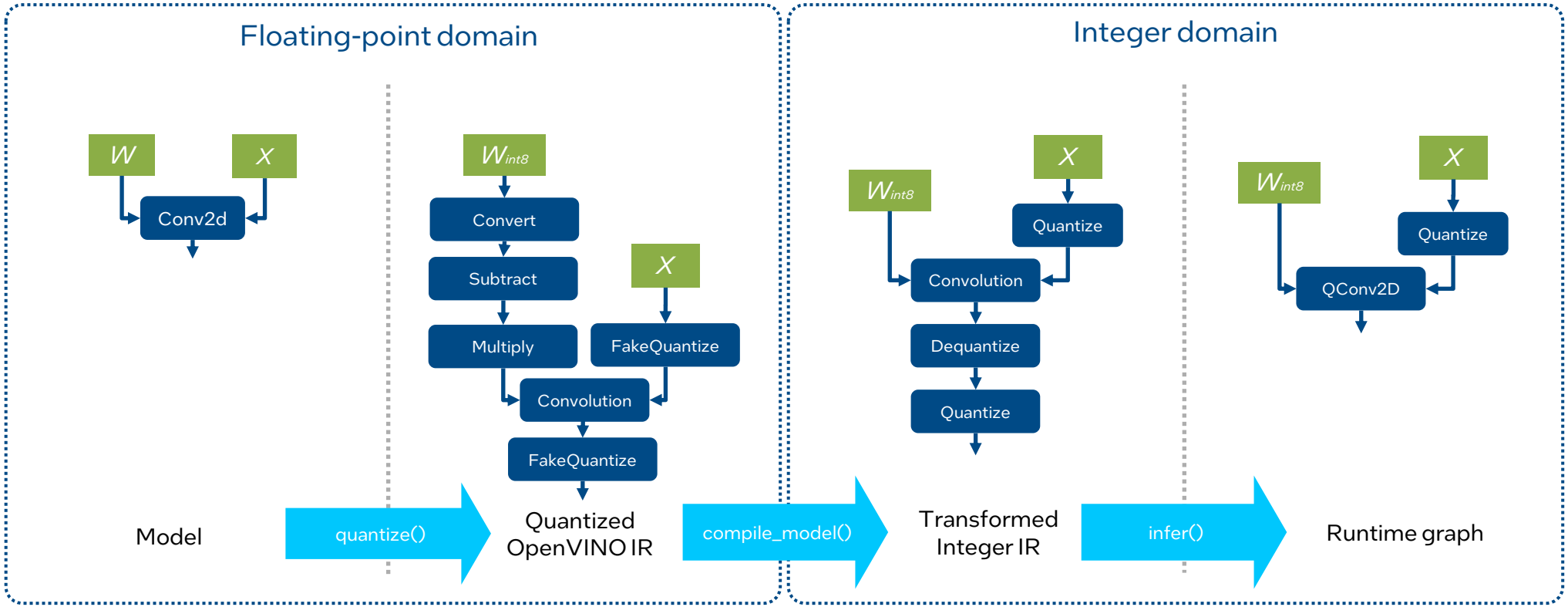

```
pytorch_model = torchvision.models.resnet50(pretrained=True)

ov_model = ov.convert_model(pytorch_model,
                            input={"input1": [1,3,224,224]})

ov.save_model(ov_model, "converted_model.xml")
```

模型量化（压缩）




```
int8_value = round(real_value / scale) + zero_point
```

Stages of the Model (OpenVINO + NNCF)



W = weights

X = input

模型压缩

OpenVINO™工具套件的神经网络压缩框架

量化路径

权重压缩

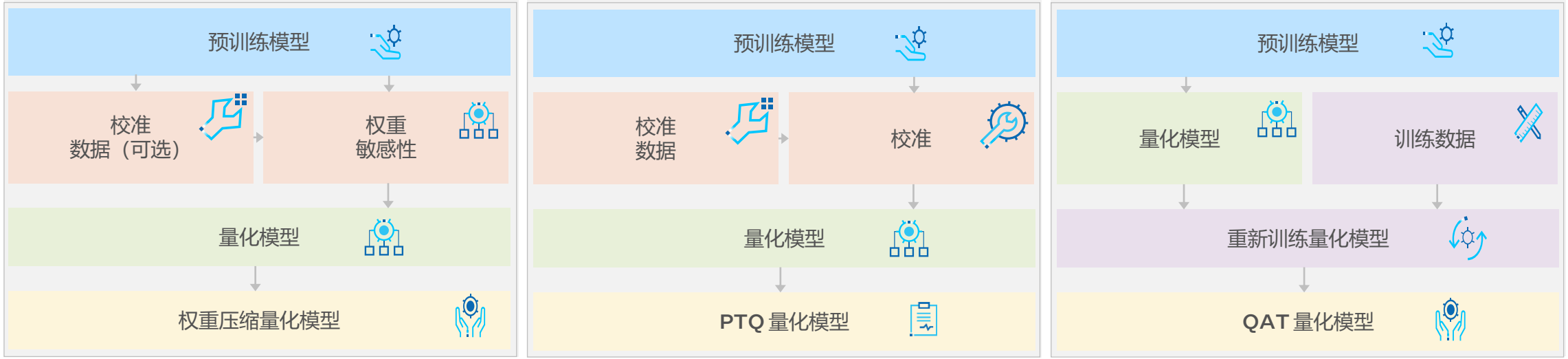
一种易于使用的量化方法，用于减少大型语言模型的占用空间和推理加速，提供 int8、int4 和 nf4 量化模型大小。

后训练量化(PTQ)

在训练后量化预训练的模型，降低模型精度以提高内存使用率和推理速度，同时可能牺牲一些准确性。

训练感知量化(QAT)

在训练过程中结合量化感知技术通过降低精度优化模型 。旨在保持准确性，但需要复杂且耗时的模型重新训练。

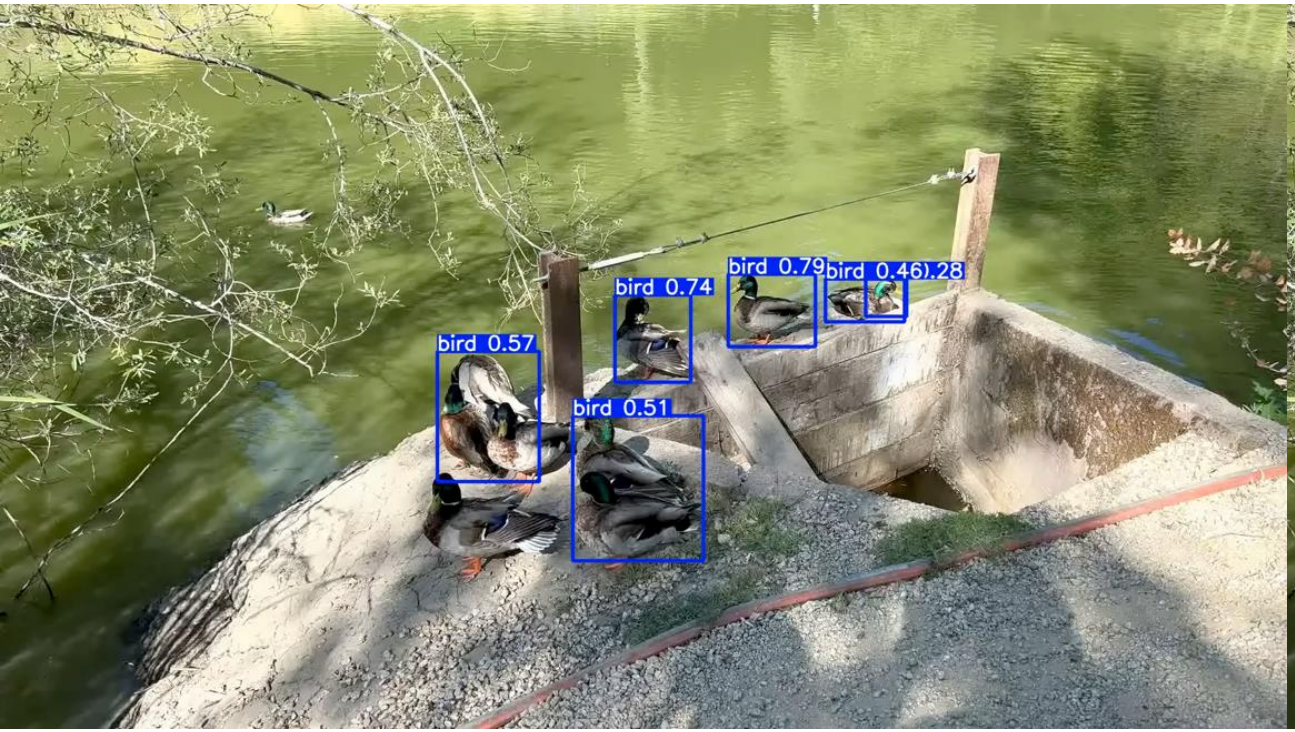


Note: Except for ONNX (.onnx model formats), all models have to be converted to an IR format to use as input to the Runtime
Development guide: https://docs.openvino.ai/2023.2/openvino_docs_model_optimization_guide.html

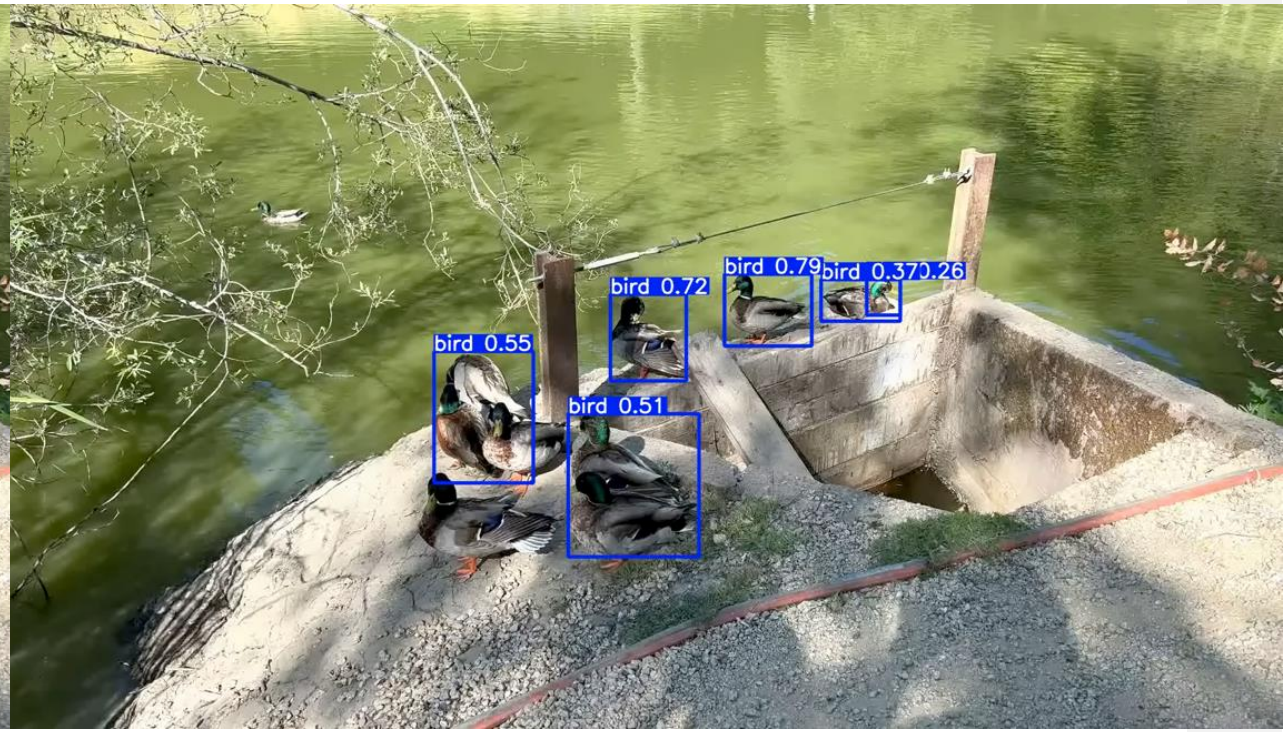
Post-Training Quantization (NNCF)

```python
from openvino import runtime as ov
import nncf

core = ov.Core()
model = core.read_model("model.xml")
data_loader = get_data_loader()

# Remove labels from data item and prepare input data (~100-500 samples)
def transform_fn(data_item):
    input_tensor = preprocess(data_item)["img"].numpy()
    return input_tensor

calibration_dataset = nncf.Dataset(data_loader, transform_fn)
quantized_model = nncf.quantize(model, calibration_dataset,
                                preset=nncf.QuantizationPreset.PERFORMANCE)
```

Quantization Results Comparison (YOLOv8)



FP32



INT8

Quantization Performance Comparison (OpenPose)

16.6 MB

4.7 MB

OpenVINO™ Runtime

Automatic Conversion

```python
from openvino import runtime as ov

img = load_img()

core = ov.Core()
# PyTorch, Tensorflow, TF Lite, ONNX, PaddlePaddle frontend
model = core.read_model(model="model.onnx")
compiled_model = core.compile_model(model=model, device_name="CPU")

output_layer = compiled_model.outputs[0]

result = compiled_model(img)[output_layer]
```

Supported Devices

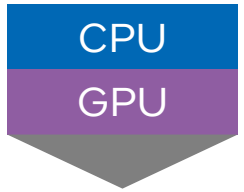
```
compiled_model = core.compile_model(model=model, device_name="CPU")
```

Supported Devices

CPU

```
compiled_model = core.compile_model(model=model, device_name="CPU")
```

Supported Devices




```
compiled_model = core.compile_model(model=model, device_name="GPU")
```

Gen AI 用例和模型支持

视频生成 · 3D建模 · 图像生成器 · 图像分割

聊天机器人 · 代码生成 · 搜索 · 文本分类 · 内容创作 · 指令遵循

音乐生成 · 文本转音频 · 音频转文本 · 单声转换

视觉

- CLIP
- BLIP
- FILM
- Pix2Pix
- Riffusion
- ControlNet
- Zero Scope
- QR code monster
- Segment Anything Model (SAM)
- Latent Consistency Models (LCM)
- Würstchen
- DeepFloyd IF
- DeciDiffusion
- Stable Diffusion

语言

- GPT J
- Notus
- LLaVa
- Llama 2/3
- BLOOM
- chatGLM
- chatGLM3
- Baichuan 2
- Neural Chat
- StableLM-Epoch-3B
- StableLM-tuned-alpha-3b
- MPT
- Dolly
- Youri
- Qwen
- Mistral
- Zephyr
- RedPajama
- Phi-3

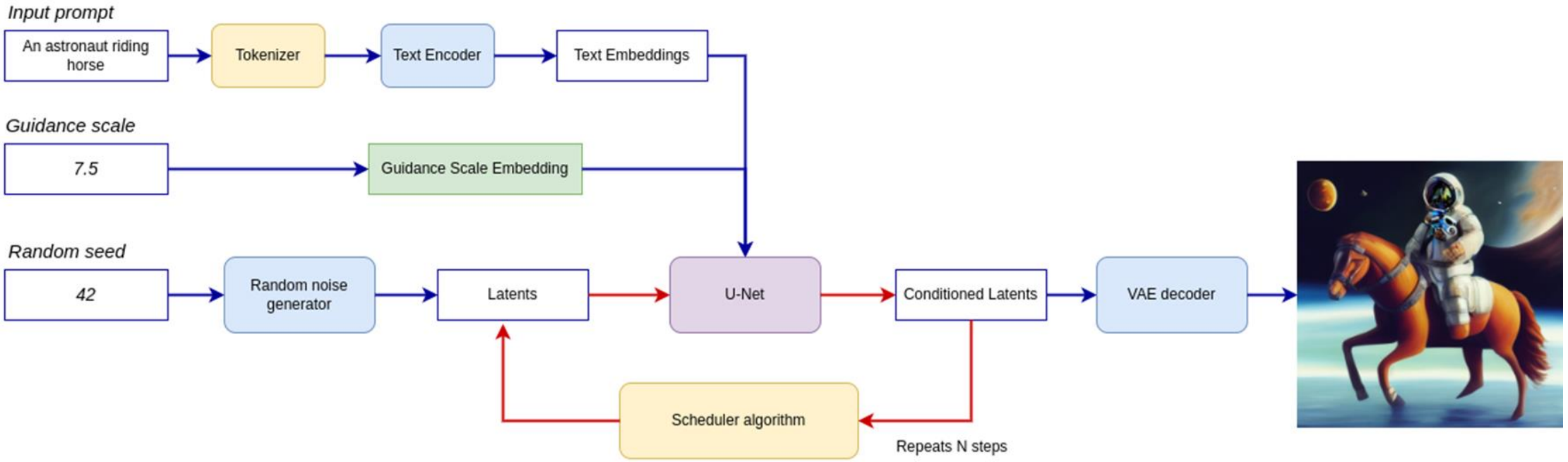
音频

- BARK
- VITS
- SoftVC
- Whisper
- MusicGen
- AudioLDM
- Distil-Whisper

示例模型支持包括，但不限于：

OpenVINO Generative AI Github* Repository介绍

GenAI任务流水线


Input prompt
An astronaut riding horse
Tokenizer
Text Encoder
Text Embeddings
Guidance scale
7.5
Guidance Scale Embedding
Random seed
42
Random noise generator
Latents
U-Net
Conditioned Latents
VAE decoder
Scheduler algorithm
Repeats N steps

GenAI模型开发流程

Optimum-Intel
(base on Transformers and Diffusers)

1 | 模型转换

PyTorch Frontend
- openvino.model_convert
- torch.compile

2 | 模型压缩

NNCF
- Weight Compression
- PTQ
- QAT

3 | 模型部署

Runtime(Backend)
- CPU
- GPU
- NPU
- ...

4 | 搭建任务流水线

- **Text-generation**
- **Text-to-image**
- **...**

GenAI模型一键导出

支持Hugging Face Remote模型的一键导出及量化

“Llama3, Phi3, ChatGLM3, Qwen, Qwen1.5, Baichuan2, InternLM, MiniCPM, StableLM, Mixtrail, Starcoder2, Orion, OLMo…”

模型转换及量化

```
optimum-cli export openvino --model <model_id_or_path> --task <task>
<out_dir> --weight-format int4
```

示例

```
optimum-cli export openvino --model meta-llama/Meta-Llama-3-8B --task text-
generation-with-past --weight-format int4 --group-size 128 --ratio 0.8 --sym
llama-3-8b-instruct/INT4_compressed_weights
```

GenAI模型开发流程

Optimum-Intel
(base on Transformers and Diffusers)

1 模型转换	2 模型压缩	3 模型部署	4 搭建任务流水线
PyTorch Frontend • openvino.model_convert • torch.compile	**NNCF** • Weight Compression • PTQ • QAT	**Runtime(Backend)** • CPU • GPU • NPU • ...	• **Text-generation** • **Text-to-image** • **...**

OpenVINO™ Integration with Optimum 加速 Diffuser

Task	Auto Class
text-to-image	OVStableDiffusionPipeline
image-to-image	OVStableDiffusionImg2ImgPipeline
inpaint	OVStableDiffusionInpaintPipeline

Python

```
- from diffusers import DiffusionPipeline
+ from optimum.intel import OVStableDiffusionXLPipeline

model_id = "stabilityai/stable-diffusion-xl-base-1.0“
- pipeline = DiffusionPipeline.from_pretrained(model_id )
+ pipeline = OVStableDiffusionXLPipeline.from_pretrained(model_id)
prompt = "train station by Caspar David Friedrich"
image = pipeline(prompt).images[0]
```

https://huggingface.co/ docs/optimum/main/en/intel/inference#diffusers-models

OpenVINO™ Integration with Optimum 加速 Transformers

Task	Auto Class
text-classification	OVModelForSequenceClassification
token-classification	OVModelForTokenClassification
question-answering	OVModelForQuestionAnswering
audio-classification	OVModelForAudioClassification
image-classification	OVModelForImageClassification
feature-extraction	OVModelForFeatureExtraction
fill-mask	OVModelForMaskedLM
text-generation	OVModelForCausalLM
text2text-generation	OVModelForSeq2SeqLM

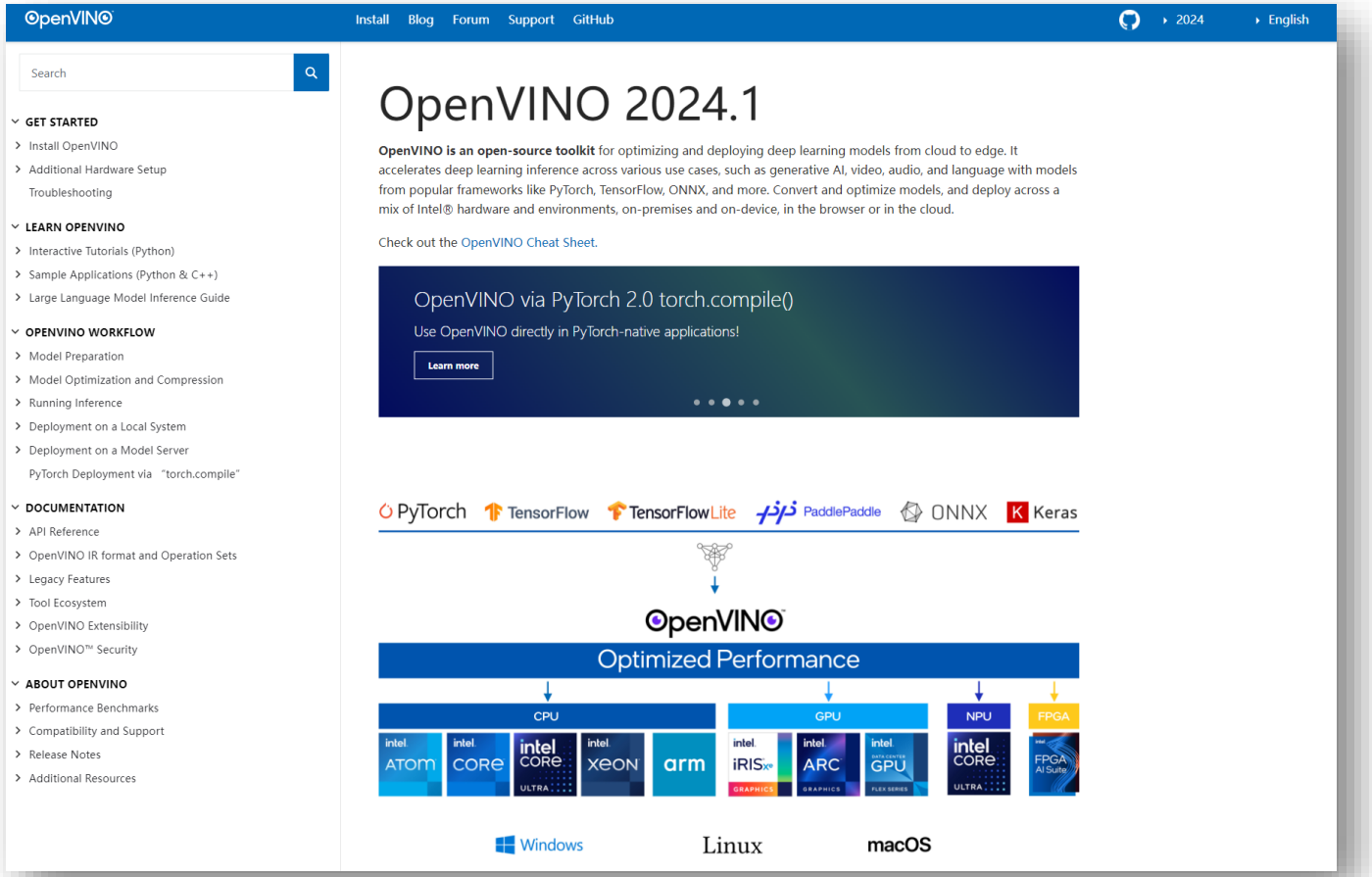
Python

```
- from transformers import AutoModelForCausalLM
+ from optimum.intel.openvino import OVModelForCausalLM

- model = AutoModelForCausalLM.from_pretrained(model_id)
+ ov_model = OVModelForCausalLM.from_pretrained(model_id)

generate_ids = ov_model.generate(input_ids)
```

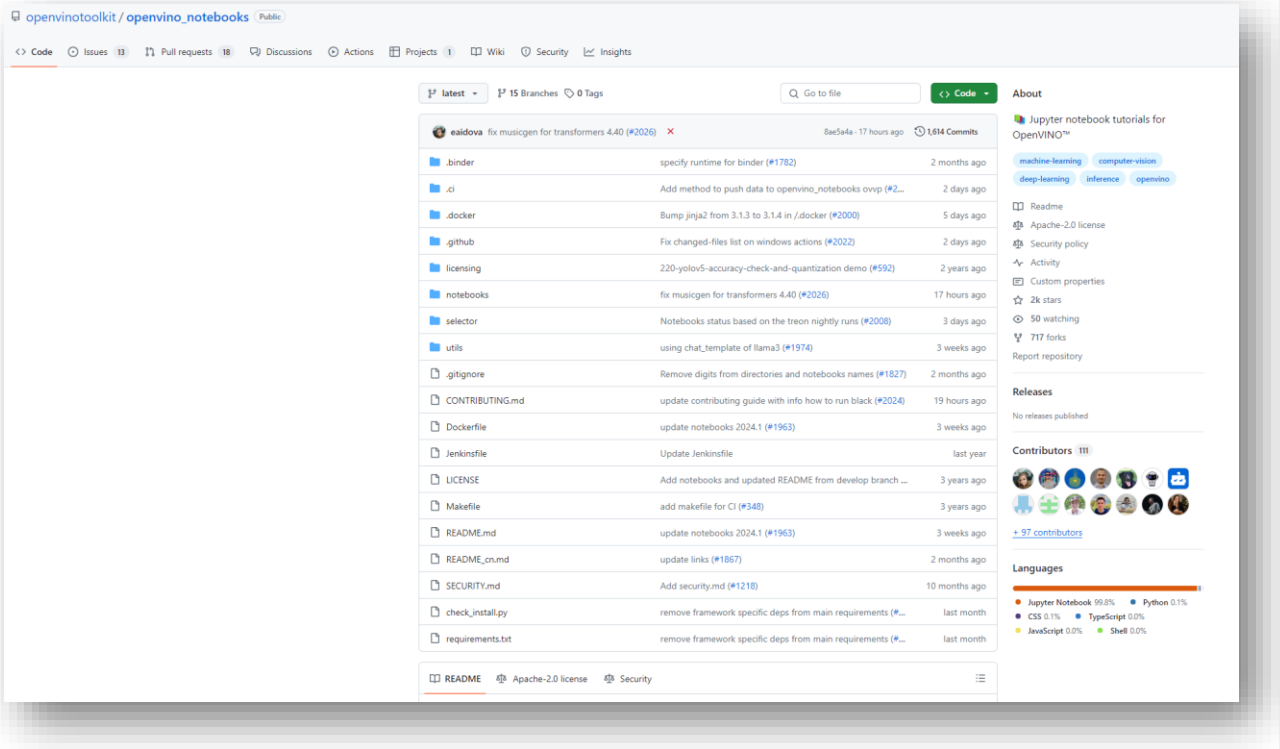
https://huggingface.co/docs/optimum/main/en/intel/inference#transformers-models

文档和示例

docs.openvino.ai



github.com/openvinotoolkit/openvino_notebooks

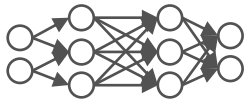
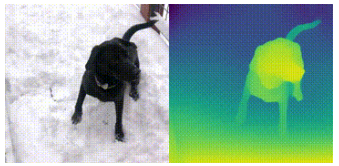


Github Jupyter Notebooks

生成式 AI、自然语言处理和计算机视觉教程

View on GitHub | Open in Colab | Launch in Binder

<Hello World/>	**Hello World**	▪ Basic introduction to OpenVINO's Python API ▪ Inference on a mobilenetv3 image classification model
{API}	**OpenVINO API**	▪ Load Runtime and Show Info ▪ Loading a Model ▪ Getting Information about a Model ▪ Doing Inference on a Model ▪ Reshaping and Resizing
TensorFlow	**Tensorflow to OpenVINO**	▪ Demonstrates how to convert TensorFlow models to OpenVINO IR
PyTorch	**PyTorch to OpenVINO**	▪ Convert PyTorch models to OpenVINO IR ▪ Uses Model Optimizer to convert the open source fastseg semantic segmentation model

TensorFlow 	**Tensorflow Training**	▪ End-to-end training to deployment workflow starting with TensorFlow's Flowers classification demo
Model Optimizer & Runtime	**Model Tools**	▪ Download a model, ▪ Convert to OpenVINO's IR format ▪ Get model information, ▪ Benchmark model
	**Mono-Depth**	▪ Demonstrates Monocular Depth Estimation with MidasNet model ▪ Users can upload their own videos and images, input data will be resized
	**Background Removal**	▪ Background removal in images ▪ The open source U^2-Net model is converted from PyTorch

GenAI 开发人员教程

使用GenAI 和 LLM Jupyter Notebooks 启动您的 AI PC 应用程序

Large Language Model (LLM) Chatbot	▪ 使用 OpenVINO 工具套件制作由 LLM 提供支持的聊天机器人
Stable Diffusion* v2	▪ 利用 Stable Diffusion* v2 和 OpenVINO 工具套件探索文本到图像生成和无限缩放功能
LLM Instruction Following	▪ 运行遵循指令的文本生成流水线
Bootstrapping Language-Image Pretraining (BLIP)	▪ 将 BLIP 用于视觉语言处理任务，例如视觉问答和图像字幕
Latent Consistency Models (LCM)	▪ 了解如何使用 LCM 和 OpenVINO 工具套件生成图像.
MusicGen	▪ 了解单级自回归转换器模型，该模型可根据文本描述或音频提示生成高质量的音乐样本
Distil-Whisper Model	▪ 使用此模型和 OpenVINO 工具套件体验自动语音识别.
YOLOv8* Optimization	▪ 了解如何转换和优化 YOLOv8* 模型.

下载HuggingFace模型的方法

方法一：设置镜像网站

CLI

```
pip install -U huggingface_hub
export HF_ENDPOINT=https://hf-mirror.com
```

方法二：本地下载

CLI

```
git clone https://www.modelscope.cn/qwen/Qwen-1_8B-Chat.git
```

CLI

```
optimum-cli export openvino --model Qwen-1_8B-Chat ...
```

Demo分享