非常好 👍 Feng Jun，这一步非常关键——**配置完整的GPU深度学习环境**，是你后面所有AI训练与ONNX推理实验的基础。下面我会一步步帮你搭好环境（以 Windows 10/11 + NVIDIA GPU 为例；如果你用Linux我也可以给另一份方案）。

🧩 一、环境安装目标（最终结果）

安装完成后，你的环境将具备：

组件	版本建议	功能说明
Python 3.10 / 3.11	✅ 推荐3.10	稳定兼容各深度学习库
CUDA Toolkit 12.1+	✅ 推荐12.1	GPU计算核心环境
cuDNN 8.x	✅ 与CUDA匹配	深度学习GPU加速库
PyTorch (GPU版)	✅ 自动检测CUDA	用于模型训练与导出
ONNX / ONNX Runtime	✅ CPU+GPU支持	模型导出与推理执行
Netron	✅ 可视化工具	查看ONNX模型结构
Jupyter / VS Code	✅ IDE	编写与调试代码

⚙️ 二、安装准备

在开始前请确认：

1. 有 **NVIDIA 显卡**（GTX 10系列及以上）。
2. 安装了 **NVIDIA驱动程序**（版本 ≥ 531.79）。
    - 右键桌面 → NVIDIA控制面板 → 帮助 → 系统信息，查看“驱动版本”。
3. 电脑磁盘留有 ≥ 10 GB空间。

🪄 三、安装步骤详解（Windows 版）

① 安装 Anaconda（推荐）

Anaconda 可以帮你快速管理Python环境和库，避免版本冲突。

1. 打开下载页：
   🔗 https://www.anaconda.com/download
2. 选择 Windows → Python 3.10 版本。
3. 安装时勾选：
   - ✅ “Add Anaconda to PATH”（方便命令行使用）
   - ✅ “Register Anaconda as my default Python”

安装完成后，打开 **Anaconda Prompt** 或 **PowerShell**。

② 创建虚拟环境

建议单独建一个专用于AI学习的环境：

```
conda create -n ai_env python=3.10
conda activate ai_env
```

③ 安装 PyTorch（自动安装CUDA/cuDNN）

PyTorch会自动安装兼容的CUDA与cuDNN版本，因此**无需手动装CUDA Toolkit**。

访问官网查看命令：
🔗 https://pytorch.org/get-started/locally/

选择：

- PyTorch Build: Stable
- Your OS: Windows
- Package: Conda
- Language: Python
- Compute Platform: CUDA 12.1

命令如下：

```
conda install pytorch torchvision torchaudio pytorch-cuda=12.1 -c pytorch -c nvidia
```

安装过程会自动下载：

- CUDA 12.1 runtime
- cuDNN
- PyTorch核心库

④ 验证 PyTorch GPU 是否可用

进入Python交互环境：

```
python
```

输入以下命令：

```
import torch
print(torch.__version__)          # 查看PyTorch版本
print(torch.cuda.is_available())  # True表示CUDA可用
print(torch.cuda.get_device_name(0))  # 输出显卡型号
```

✅ 如果返回 True 且打印出显卡名称（如 RTX 3060），说明GPU环境成功！

⑤ 安装 ONNX 与 ONNX Runtime

```
pip install onnx onnxruntime-gpu
```

验证：

```
import onnxruntime as ort
print(ort.get_device())
```

输出应为：

GPU

⑥ 安装可视化与辅助工具

```
pip install netron jupyter matplotlib numpy opencv-python tqdm
```

说明：

- **Netron**：ONNX模型可视化
- **Jupyter**：交互式笔记本环境
- **matplotlib/numpy**：基础数值计算与绘图
- **opencv-python**：图像读写与预处理
- **tqdm**：训练进度条

启动 Jupyter：

```
jupyter notebook
```

或使用 VS Code + Python 插件。

⑦（可选）安装 CUDA Toolkit & cuDNN（手动方式）

如果你想自己配置或用于其他框架（如 TensorFlow），可以额外安装：

1. CUDA Toolkit 12.1：
   🔗 https://developer.nvidia.com/cuda-downloads
2. cuDNN 8.9+：
   🔗 https://developer.nvidia.com/cudnn

安装完后将其 `bin/`，`lib/`，`include/` 路径加入 环境变量。
但**PyTorch用户一般不需要手动这一步**。

🧠 四、测试代码（确认环境完整性）

```
import torch
import torchvision.models as models
import onnx
import onnxruntime as ort
import numpy as np

# 1. 测试PyTorch GPU训练
print("GPU available:", torch.cuda.is_available())

# 2. 载入预训练模型
model = models.resnet18(pretrained=True).eval().cuda()

# 3. 导出ONNX
x = torch.randn(1, 3, 224, 224).cuda()
torch.onnx.export(model, x, "resnet18.onnx")

# 4. 加载并在ONNX Runtime运行
session = ort.InferenceSession("resnet18.onnx", providers=['CUDAExecutionProvider'])
print("ONNX Runtime device:", session.get_providers())

# 5. 运行推理
output = session.run(None, {"input.1": np.random.randn(1, 3, 224, 224).astype(np.float32)})
print("Output shape:", output[0].shape)
```

若能正常运行且输出结果形状正确，说明你的**AI训练 + 推理 + GPU环境完全就绪！**

📘 五、可选扩展：后续推荐工具

工具	用途
Netron (桌面版)	双击 ONNX 文件即可查看网络结构
TensorBoard	监控训练过程
VS Code	主力IDE，集成Jupyter内核
NVIDIA Nsight Systems	进行GPU性能分析

是否需要我再帮你把这个安装流程写成一个**自动化一键脚本（.bat 文件）**？
运行一次就能自动创建 conda 环境并安装全部库。