为OpenFold推理建立独立环境并用小样例验收

完整操作教程 · 参考固定 · 往返恢复验收

OpenFold推理环境应与训练环境分开记录和验收。训练需要的数据、优化器或分布式配置，并不自动适用于预测任务；同样，import成功也不能证明CUDA扩展、参数和输入数据匹配。本篇采用Linux下官方当前安装路线，用一个单体和预计算比对完成推理验收。

运行边界：示例代码需按安装版本和真实项目输入复核；未提供真实生物数据，语法或小样本检查不代表科学结果和性能验收。

OpenFold推理环境应与训练环境分开记录和验收。训练需要的数据、优化器或分布式配置，并不自动适用于预测任务；同样，import成功也不能证明CUDA扩展、参数和输入数据匹配。本篇采用Linux下官方当前安装路线，用一个单体和预计算比对完成推理验收。

先固定兼容组合和运行范围

官方当前安装说明面向CUDA 12与PyTorch 2，具体依赖以选定commit的environment.yml为准，不能仅记这两个大版本。先确认驱动、CUDA工具链与编译器兼容，再建立独立环境。这里不提供适合任意显卡的统一显存规格，也不直接把现有训练环境的二进制扩展复制过来。

材料	本例要求	验收依据
代码与环境	固定commit及独立Conda环境	安装日志、环境导出与扩展测试
参数	model_1_ptm对应AlphaFold参数	来源、校验与config_preset匹配
序列与MSA	一个FASTA、对应预计算比对目录	query ID与查询序列一致
模板目录	匹配的mmCIF材料	即使某路线无模板仍需按接口提供目录

建立独立安装记录

先初始化Conda/Mamba的Linux shell，确认openfold_infer_001环境名尚未使用。代码固定本篇已核对的官方commit，升级时新建环境复测。下例会创建新环境并编译组件，安装前阅读该commit的官方脚本；完成后保留全部日志。这里是教程操作，不能将命令列出等同于已经替读者部署。

```
set -euo pipefail
OPENFOLD_COMMIT=be2ec1841f16c966c65ae0e7599ebbadc725757d
[[ "$OPENFOLD_COMMIT" =~ ^[0-9a-f]{40}$ ]]
git clone https://github.com/aqlaboratory/openfold.git openfold-infer-001
cd openfold-infer-001
git checkout --detach "$OPENFOLD_COMMIT"
git rev-parse HEAD > source-commit.txt
mamba env create -n openfold_infer_001 -f environment.yml > environment-create.log 2>&1
conda activate openfold_infer_001
bash -e scripts/install_third_party_dependencies.sh > third-party-install.log 2>&1
conda deactivate
conda activate openfold_infer_001
export LIBRARY_PATH="$CONDA_PREFIX/lib:${LIBRARY_PATH:-}"
export LD_LIBRARY_PATH="$CONDA_PREFIX/lib:${LD_LIBRARY_PATH:-}"
python -c 'import torch,openfold; print(torch.__version__,torch.version.cuda); assert
torch.cuda.is_available()' \
  > environment-check.txt
conda env export > environment-resolved.yml
python run_pretrained_openfold.py --help > inference-help.txt
```

使用一份匹配样例完成预测

继续在仓库目录和已激活环境执行。FASTA_DIR只含一个已核对FASTA；ALIGNMENTS按官方单体样例布局，每个query ID一子目录，包含所需a3m、sto和hhr等文件。PARAMS是适用的params_model_1_ptm.npz，不混用OpenFold训练checkpoint参数入口。TEMPLATES为对应模板mmCIF目录。

```
set -euo pipefail
: "${FASTA_DIR:?■■■FASTA■■}" "${ALIGNMENTS:?■■■■■■■■■■}"
: "${TEMPLATES:?■■mmCIF■■}" "${PARAMS:?■■model_1_ptm■npz■■}"
test -d "$FASTA_DIR"
test -d "$ALIGNMENTS"
test -d "$TEMPLATES"
test -s "$PARAMS"
mkdir inference-smoke-001
sha256sum "$PARAMS" > inference-smoke-001/params.sha256
/usr/bin/time -v -o inference-smoke-001/run.time python run_pretrained_openfold.py \
  "$FASTA_DIR" "$TEMPLATES" --use_precomputed_alignments "$ALIGNMENTS" \
  --config_preset model_1_ptm --jax_param_path "$PARAMS" --model_device cuda:0 \
  --data_random_seed 101 --skip_relaxation --output_dir "$PWD/inference-smoke-001" \
  > inference-smoke-001/run.log 2>&1
mapfile -d '' -t models < <(find inference-smoke-001 -type f -name '*_unrelaxed.pdb' -
print0)
test "${#models[@]}" -eq 1
test -s "${models[0]}"
```

验收环境、输入和结果三件事

读取日志确认参数加载、特征处理和模型推理均完成，再解析PDB核对序列长度、残基映射及有限坐标，并检查预测置信度与模板/MSA使用情况。这个试验跳过relaxation，因此只应要求unrelaxed结果，不能报告为已完成松弛或经过实验验证的结构。

CUDA可用但扩展加载失败时，查编译日志和实际工具链；参数形状不匹配时，查preset与权重；MSA缺失时，查query目录和输入序列。不要通过切换无MSA模式绕过失败后仍称为同一预测协议。预计算比对路线也不意味着训练数据库已部署。

安装记录、数据库和参数目录分别管理，代码升级先新建环境做同一样例回归。完整试跑记录主机RSS、显存与结果路径，之后用代表长度评估批量推理；单个短序列成功只能证明这条配置在这个样例上可用，不能保证所有蛋白或多体任务的资源需求与结果可靠性。

安装脚本设置的Conda环境变量在重新激活后生效，因此代码在安装后重新激活环境。先用固定commit自带的可用单体样例验证，再替换自己的序列与匹配MSA；不能拿训练数据路径存在当推理材料已经齐备。验收表分别写环境导入、扩展可用、参数加载和完整推理四个状态。

配套练习包

本PDF对应的ZIP包含本文代码块、主题练习模板或合成数据、README和SHA256校验文件。请从同一教程页面下载练习包；先核对校验值，再在独立目录复现。

参考资料

OpenFold官方安装与兼容范围 · 来源资料

单体推理、参数与比对布局 · 来源资料

当前推理CLI实现 · 来源资料

OpenFold官方安装说明 · 来源资料

操作建议：先在合成小样本上验证流程，再依据项目参考、标签合同、存储策略与获配资源验收真实数据。