数据与参赛说明

DLA Finder Challenge

第二轮 CSST 科学数据挑战大赛 · 2026 年 3 月

1. 赛题简介

阻尼 Lyα 吸收体（Damped Lyman-Alpha absorbers，DLA）是类星体光谱中出现的强中性氢吸收结构（列密度 log N_HI ≥ 20.3），其宽阻尼翼特征会对 Lyα 森林的 BAO 测量及宇宙学参数推断产生系统性影响，需在大样本光谱中自动识别并剔除。

本赛题提供一套符合 CSST 预期观测特性的全仿真模拟 QSO 光谱数据集。参赛者需基于提供的训练集，开发自动化算法，完成对测试集中 DLA 的检测与参数回归。

> 参考实现（DESI CNN DLA Finder，可直接适配）：
> https://github.com/cosmodesi/desi-dlas/tree/realdata_kibo

2. 数据下载

训练集与测试集均存储于百度网盘，请通过以下链接下载：

> 百度网盘：CSST 数据竞赛 类星体模拟光谱
> 链接：https://pan.baidu.com/s/1NGbZkpxF-cSfHUtAX_sfyg?pwd=1234
> 提取码：1234
>
> 训练集：train.fits.gz （约 200 MB，含标签）
> 测试集：test.fits.gz （约 40 MB，不含标签）

3. 数据集说明

3.1 训练集（10,000 条光谱，含标签）

参数	说明
光谱总数	10,000 条模拟 QSO 光谱
QSO 红移	z_QSO ∈ [1.6, 4.0]
波长范围	2000 – 8000 Å（约 1842 个像素）
DLA 比例	50% 含 DLA，50% 无 DLA
每条光谱 DLA 数	0、1 或 2 个（含 DLA 的光谱中 75% 为 1 个，25% 为 2 个）
DLA 列密度	log N_HI ∈ [19.5, 22.5]，高列密度样本较少
提供内容	波长、flux、每条光谱的完整真值标签

3.2 测试集（2,000 条光谱，不含标签）

参数	说明
光谱总数	2,000 条模拟 QSO 光谱
提供内容	波长、flux、z_QSO（不含 DLA 真值，不含 HAS_DLA/Z_DLA/LOGNHI）
用途	最终评分，选手不得使用测试集训练或调参

4. FITS 文件格式

4.1 训练集文件结构

HDU 名称	类型	维度	说明
PRIMARY	PrimaryHDU	—	全局元数据 Header
WAVELENGTH	ImageHDU	[N_pix]	观测系波长数组，单位 Å, 所有光谱共用
FLUX	ImageHDU	[10000 × N_pix]	模拟观测光谱（含噪声、DLA 吸收）
FLUX_CLEAN	ImageHDU	[10000 × N_pix]	无噪声版本，用于调试验证
LABELS	BinTableHDU	[10000 行]	训练标签（见 4.3 节）

4.2 测试集文件结构

HDU 名称	类型	维度	说明
PRIMARY	PrimaryHDU	—	全局元数据 Header
WAVELENGTH	ImageHDU	[N_pix]	观测系波长数组，单位 Å

FLUX	ImageHDU	[2000 × N_pix]	模拟观测光谱（含噪声）
META	BinTableHDU	[2000 行]	仅含 TARGETID 和 Z_QSO，无 DLA 真值

4.3 LABELS 字段（训练集专有）

字段名	格式	说明
Z_QSO	float32	QSO 红移
HAS_DLA	int16	是否含 DLA：1 = 含，0 = 无
N_DLA	int16	DLA 数量（0、1 或 2）
Z_DLA1	float32	第 1 个 DLA 红移；无 DLA 时为 NaN
LOGNHI1	float32	第 1 个 DLA 的 $\log_{10}$(N_HI)，单位 log(cm⁻²)；无 DLA 时为 NaN
Z_DLA2	float32	第 2 个 DLA 红移；仅双 DLA 有效，否则为 NaN
LOGNHI2	float32	第 2 个 DLA 的 $\log_{10}$(N_HI)；仅双 DLA 有效，否则为 NaN
SNR_GU	float32	GU 波段（2550–4100 Å）逐谱信噪比（可用于分层训练）
SNR_GV	float32	GV 波段（4100–6200 Å）逐谱信噪比
SNR_GI	float32	GI 波段（6200–8000 Å）逐谱信噪比

4.4 数据读取示例

```
from astropy.io import fits
import numpy as np

# 读取训练集
with fits.open('csst_dla_train_10k.fits.gz') as hdul:
    wave   = hdul['WAVELENGTH'].data       #  (N_pix,)  单位 Angstrom
    flux   = hdul['FLUX'].data             #  (10000, N_pix)
    labels = hdul['LABELS'].data           # BinTable

z_qso   = labels['Z_QSO']
has_dla = labels['HAS_DLA'].astype(bool)  # True / False
z_dla1  = labels['Z_DLA1']                # NaN if no DLA
lognhi1 = labels['LOGNHI1']               # NaN if no DLA
snr_gu  = labels['SNR_GU']                # per-spectrum SNR

# 读取测试集
with fits.open('csst_dla_test_2k.fits.gz') as hdul:
    wave_t = hdul['WAVELENGTH'].data       #  (N_pix,)
    flux_t = hdul['FLUX'].data             #  (2000, N_pix)
    meta   = hdul['META'].data
    targetid = meta['TARGETID']            # 光谱唯一编号
    z_qso_t  = meta['Z_QSO']               # QSO 红移
```

5. 参赛任务

参赛者需基于训练集训练模型，并对测试集中的每条光谱完成以下三项任务：

	任务	具体要求
①	**DLA 分类检测**	判断每条光谱是否存在 DLA，输出分类结果（含/不含）及置信度
②	**红移回归**	对每个检测到的 DLA 输出吸收体红移 z_DLA 及其不确定度
③	**列密度回归**	对每个检测到的 DLA 输出列密度 log N_HI 及其不确定度

> 一条光谱中可能存在 0、1 或 2 个 DLA，提交结果时每个 DLA 单独占一行。
> 若判断某条光谱无 DLA，则该光谱不在结果文件中出现（或以 CONFIDENCE=0 提交）。

6. 提交内容与格式

参赛者需提交以下两项内容：

6.1 预测结果文件（必须）

在测试集 FITS 文件基础上追加预测结果，生成新的 FITS 文件提交。新文件须包含测试集原有的所有扩展（WAVELENGTH、FLUX、META），并新增一个 RESULTS 扩展，格式如下：

字段名	格式	说明
TARGETID	int64	对应测试集 META 中的光谱唯一编号（可重复，一行对应一个 DLA）
Z_QSO	float32	QSO 红移（直接从 META 转填）
Z_DLA	float32	预测的 DLA 吸收体红移
LOG_NHI	float32	预测的 $\log_{10}$(N_HI)，单位 log(cm^{-2})
CONFIDENCE	float32	检测置信度，范围 [0, 1]

```
# 提交示例：在测试集 FITS 基础上追加 RESULTS 扩展
from astropy.io import fits
from astropy.table import Table
import numpy as np, shutil

# 复制测试集文件作为基础
shutil.copy('csst_dla_test_2k.fits.gz', 'my_submission.fits')

# 构建预测结果表（每行对应一个 DLA）
results = Table({
    'TARGETID':   np.array([0, 0, 5, 12, ...], dtype=np.int64),
    'Z_QSO':      np.array([2.31, 2.31, 1.95, 3.10, ...], dtype=np.float32),
    'Z_DLA':      np.array([2.10, 2.06, 1.80, 2.97, ...], dtype=np.float32),
```

```
    'LOG_NHI':    np.array([21.2, 20.8, 20.5, 21.6,  ...], dtype=np.float32),
    'CONFIDENCE': np.array([0.98, 0.85, 0.90, 0.77,  ...], dtype=np.float32),
})

# 追加到 FITS 文件
with fits.open('my_submission.fits', mode='append') as hdul:
    hdul.append(fits.BinTableHDU(results.as_array(), name='RESULTS'))
```

6.2 模型代码与说明（必须）

参赛者还须提交以下内容，统一打包为 zip 文件：

- **训练代码：** 完整可运行的训练脚本，包含数据加载、模型定义、训练循环及推理代码
- **模型介绍文档：** 简要说明所用模型架构、训练策略、预处理方式及关键超参数（1–3 页 PDF 或 Markdown）
- **预训练权重（可选）：** 若提供模型权重文件，须确保提交的结果文件可由该权重复现

> 提交文件命名建议：
> 预测结果： [队伍名]_submission.fits
> 代码与说明：[队伍名]_code.zip

7. 评分规则

最终总分由三部分加权构成：

> Final Score = 0.6 × 检出得分 + 0.4 × 参数精度得分
>
> 参数精度得分 = 加权平均（红移精度子分 + 列密度精度子分）

7.1 匹配规则

- 预测 DLA 与真值 DLA 按速度差匹配：$\Delta v = c \times |z_{pred} - z_{true}| / (1 + z_{true})$
- $\Delta v < 600$ km/s 视为匹配成功；一对多或多对一时保留速度差最小的一组。

7.2 检出得分（占总分 60%）

- 按（SNR × log N_HI）二维区间分 bin，在每个 bin 内计算 Completeness 和 Purity。
- F1 = 2 × Completeness × Purity / (Completeness + Purity)
- 所有 bin 的 F1 按该 bin 真值 DLA 数量加权平均，得到检出总分。

7.3 参数精度得分（占总分 40%）

- 仅对成功匹配的 DLA 计算，统计 Δv（红移误差）和 Δlog N_HI 的 bias 与 scatter。
- 误差越小、偏差越接近 0，得分越高。

- ERR_Z 和 ERR_LOGNHI 应与实际误差分布一致（68% 置信区间应覆盖 68% 样本）。

8. 参考实现

主办方提供基于 DESI 数据开发的 CNN DLA Finder 作为参考实现，参赛者可直接参考其架构，结合本数据集的特点进行适配：

仓库地址：https://github.com/cosmodesi/desi-dlas
推荐分支：realdata_kibo

包含内容：
- 1D CNN 模型定义（PyTorch）
- FITS 数据加载器
- 多任务训练脚本（检测 + z 回归 + log N_HI 回归）
- 推理与 catalog 生成脚本

建议：训练时充分利用 SNR_GU 字段进行分层采样，确保低 SNR 样本（SNR_GU < 0.5）有足够代表性，以提升低信噪比条件下的检出完备性。