### 基于飞桨实现乒乓球时序动作定位大赛 ：B榜第4名方案

1. 模型思路

**思路步骤：**

1. 首先，对训练集进行特征提取。考虑定位动作的时间非常短，为了可以让模型更好的识别动作，以固定长度4秒分割动作，以每4s进行分割（例如1-4，4-8...依次进行），将这些固定长度的动作片段中标注成.npy文件。这是按照真是动作的开始时间结束时间进行分割，保证分割片段中都是完整的动作。

2.接着，在训练BMN模型前需矫正标签和数据是否一一对应，删除一些无对应标签的特征。

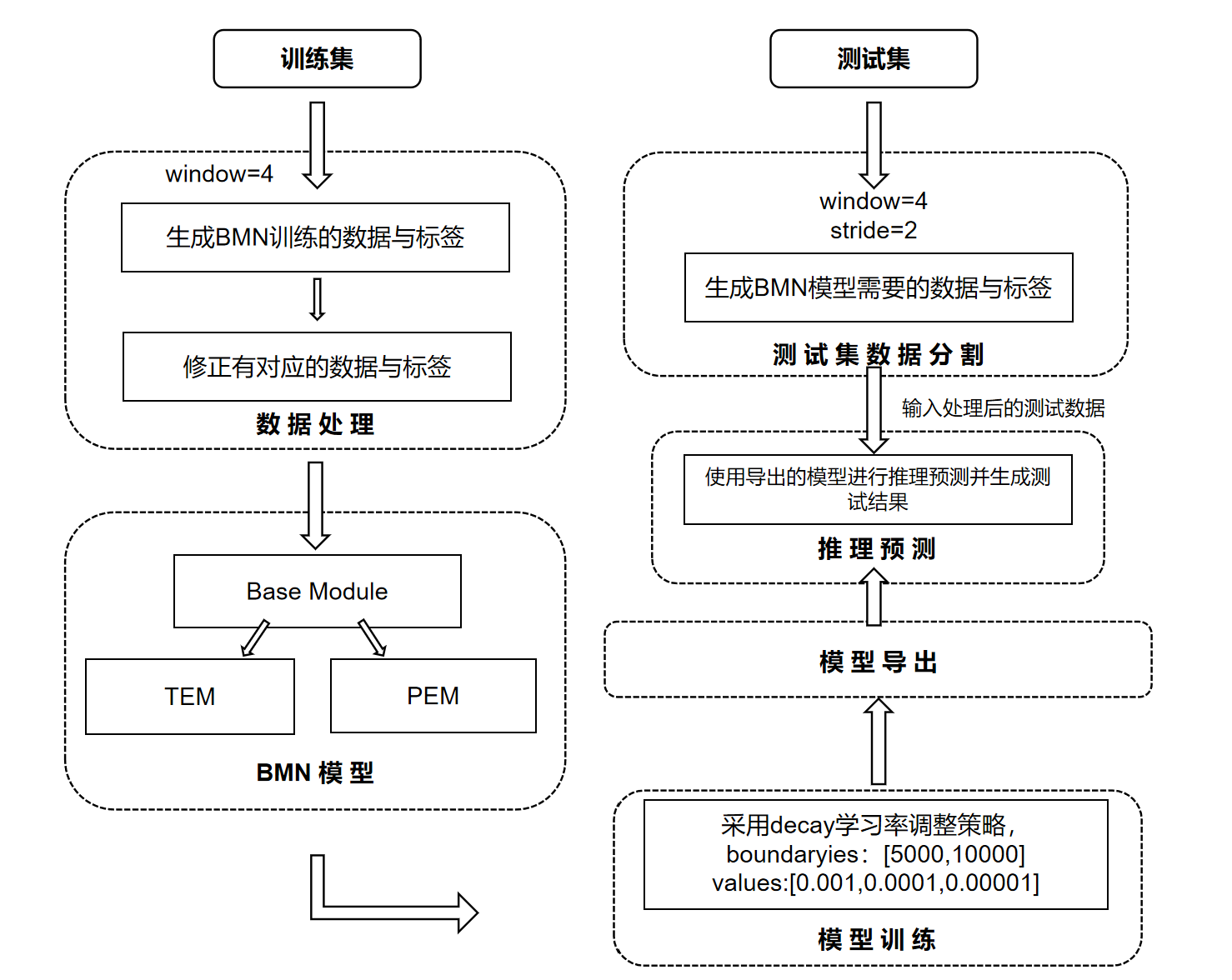
3.然后，在模型构建上，本方案采用PaddleVideo中的BMN模型。此模型引入边界匹配。(Boundary-Matching, BM)机制来评估proposal的置信度，按照proposal开始边界的位置及其长度将所有可能存在的proposal组合成一个二维的BM置信度图，图中每个点的数值代表其所对应的proposal的置信度分数。网络由三个模块组成，基础模块作为主干网络处理输入的特征序列，TEM模块预测每一个时序位置属于动作开始、动作结束的概率，PEM模块生成BM置信度图。

4.在模型训练上，最终提交结果是采用decay的学习率调整策略，设置boundaries为[5000,10000],values为[0.001,0.0001,0.00001],且训练15个epoch。导出第15个epoch的模型参数用于推理预测。

4.在测试集的数据处理上，基于训练集分割方法，将滑窗移动步长改为2，使每次分割都有2秒的重叠，保证了动作分割的完整性，能够有效提高模型精度。

5.最后，用导出的模型参数应用于处理好的测试集的特征，得到测试结果。

**整体架构图：**



1. 数据分割

由于乒乓球每个动作时间非常短，为了让模型更好的识别动作，因此这里需要将将数据进行分割。

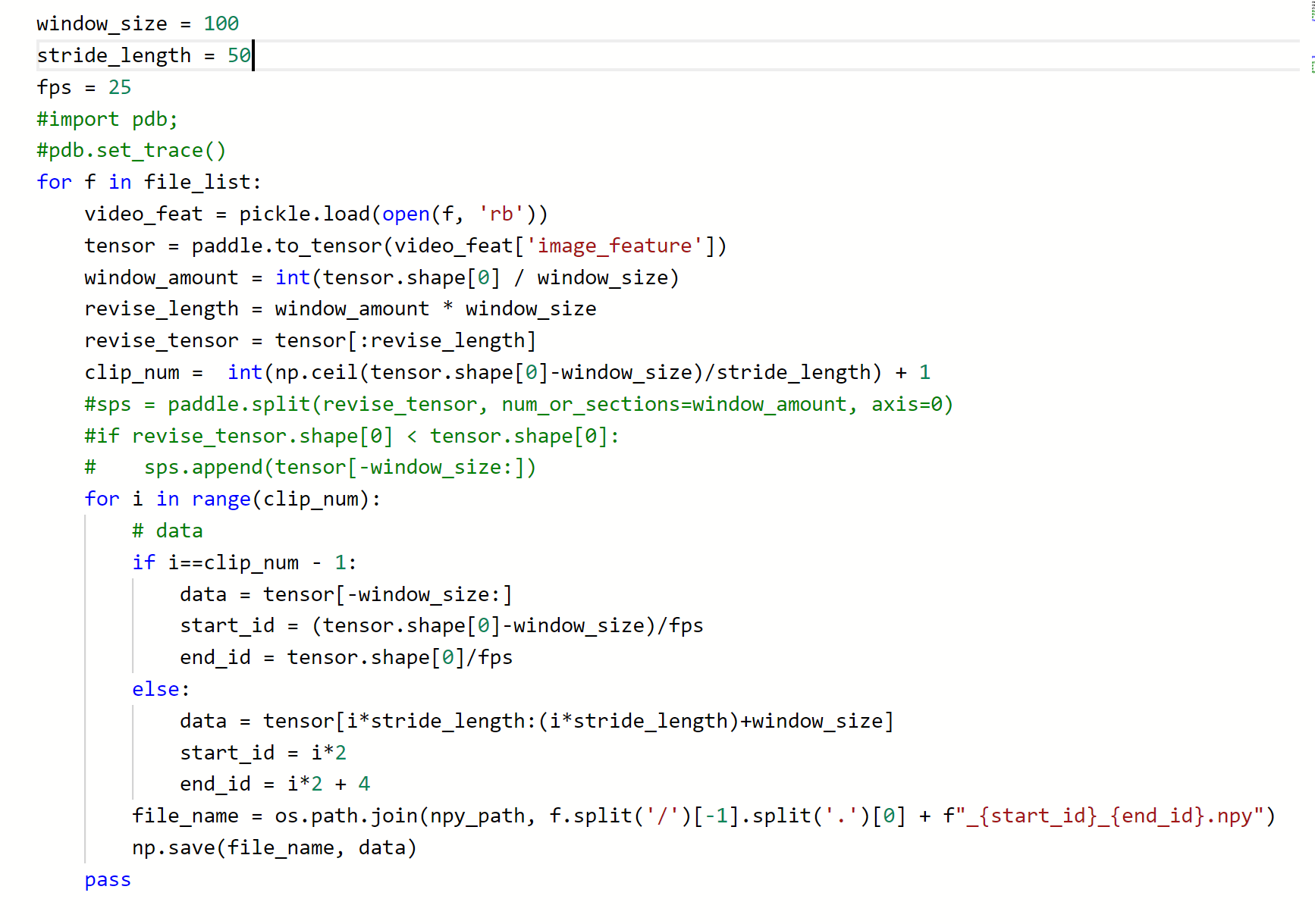
**对于训练集**，设置window=4，使动作长度统一为4s，不固定滑窗步长，每一次跳至下一个文件进行划分，然后将这些固定长度的视频片段中的每一个动作标注成.npy文件，并将其中的动作开始和结束时间变成以零为起点的基准。

# 生成bmn训练数据和标签

%cd /home/aistudio/PaddleVideo/applications/TableTennis/

!python3.7 get\_instance\_for\_bmn.py

**对于测试集**，依旧保持动作长度统一为4s，但单个动作长度为0~2s，这样可能会产生一个动作划分到两个文件的情况。考虑动作的完整性，设置滑动窗口的滑动步长stride为2，使每次划分有2秒的重叠，因此在这2秒的重叠中必然包括了某个完整动作。这样分割方法的改进，使动作特征数据更加完整，匹配精度有所提高。



#划分测试集

%cd /home/aistudio/PaddleVideo/applications/TableTennis/

# split\_testdata\_overlap.py 文件中的路径可能需要修改

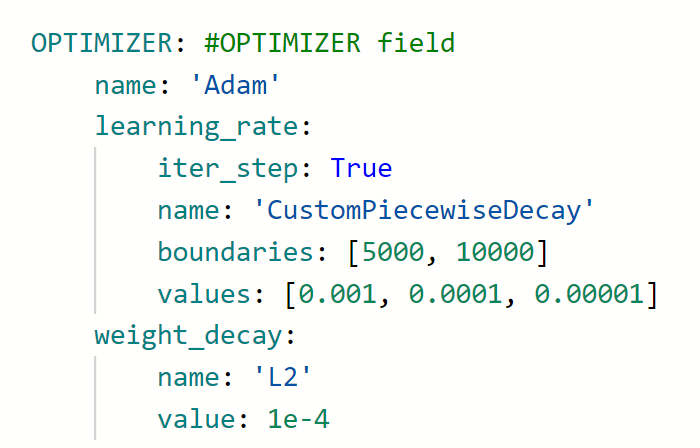
!python split\_testdata\_overlap.py

1. 调参优化策略

我们尝试了decay、warmup、average checkpoints三种学习率调整策略，分别如下：

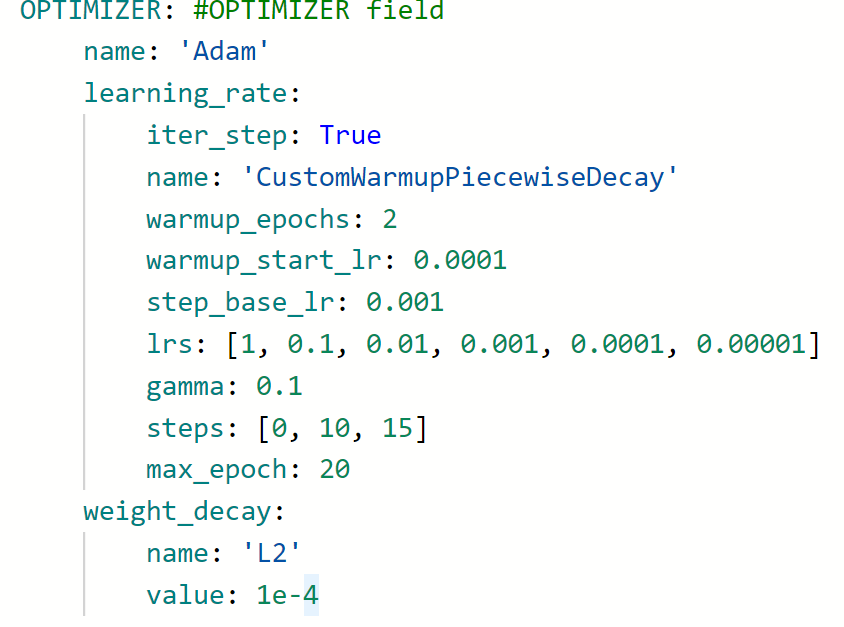
Decay 策略

**learningrate：[0.001->0.0001->0.00001]**：

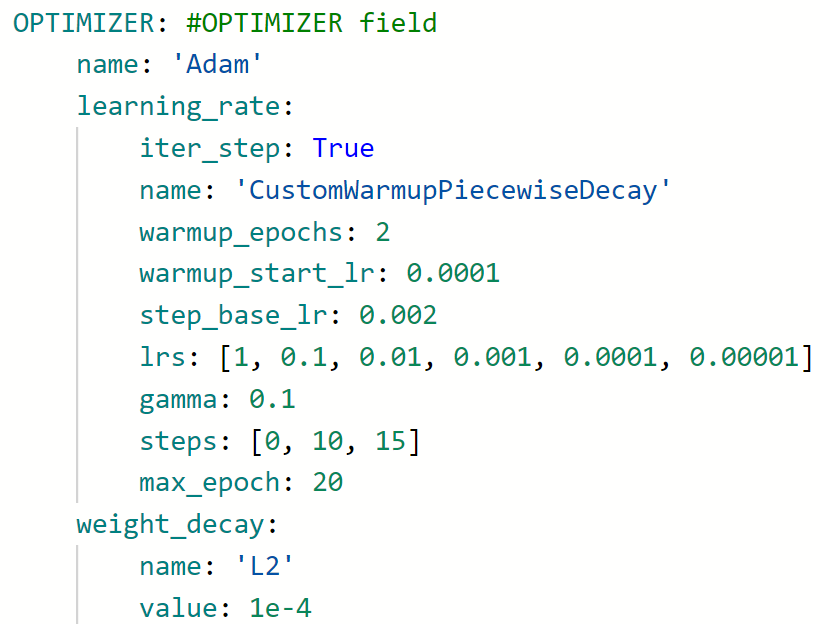


Warmup 策略：

**max\_learningrate=[0.001]**



**max\_learningrate=[0.002]**



Average checkpoints 策略：

对最后训练的5个epoch的模型参数求平均形成新的pdparams文件，用于测试集推理预测。



1. 训练脚本/代码

在不同的train step范围设定递减的学习率，使模型训练更加收敛。

resume\_epoch可以设置为指定的checkpoint并继续训练，训练后权重会保存在output文件夹中。

#训练模型

%cd /home/aistudio/PaddleVideo/

!python main.py -c configs/localization/bmn.yaml

训练日志：



1. 测试脚本/代码

#生成测试结果

%cd /home/aistudio/PaddleVideo/

!python tools/predict.py --input\_file /home/aistudio/work/data/test\_npy \

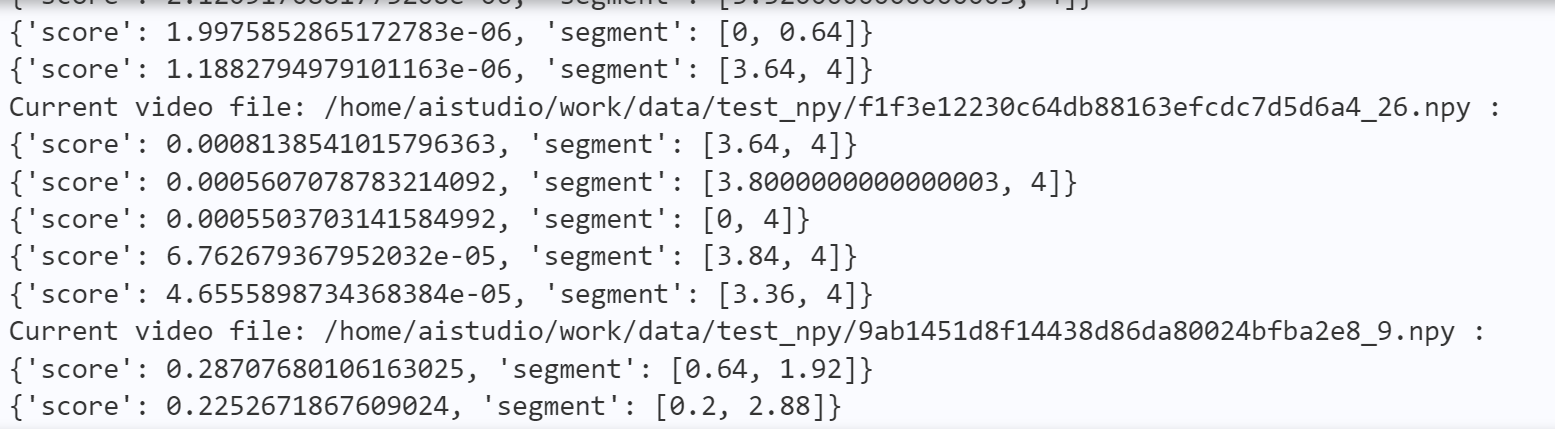
 --config configs/localization/bmn.yaml \

 --model\_file inference/BMN/BMN.pdmodel \

 --params\_file inference/BMN/BMN.pdiparams \

 --use\_gpu=True \

 --use\_tensorrt=False



上面程序输出的json文件是分割后的预测结果，运用非极大抑制算法将这些文件组合到一起，形成submission.json文件。

#合并生成结果

%cd /home/aistudio/PaddleVideo/applications/TableTennis/

!python merge\_result.py

1. 提升思路

1.在官方提供的baseline基础上增加训练的epoch数量

2.采用多步长衰减与Warmup的余弦退火衰减等多种学习率调整方法，其A榜测试提交得分对比如下：

|  |  |
| --- | --- |
| **学习率调整策略** | **15 epoch** |
| **decay 策略（learningrate：[0.001->0.0001->0.00001]）** | 44.83078 |
| **warm up策略（max\_learningrate=[0.001]）** | 44.76696 |
| **warm up策略（max\_learningrate=[0.002]）** | 44.52029 |
| **average checkpoint策略（最近5个epoch）** | 44.22303 |

可见其多种学习率调整策略对该数据结果相差不大。

1. 基于每4秒一分的划分方法上，将滑动窗口的步长改为2秒，保证测试集动作的完整性，优化特征数据。
2. 采用非极大抑制算法合并测试结果
3. 模型开发过程、训练技巧、创新思路等其它需要说明的内容

1）在运行前请校对各运行脚本中的数据路径，以及configs/localzation/bmn.yaml文件中的数据路径

2）第一次解压处理训练数据后，以及将处理好的特征储存在work/data/Input\_Feature中，第二次训练模型时可以无需再解压分割训练数据

3）若运行时间过长，模型训练中断；下一次训练时，可将bmn.yaml文件中的resume\_epoch设置为上一次中断的epoch数，模型将从读取该epoch参数继续训练。这样可以避免每次训练从头来过，耗费太多时间。

4）训练得到模型参数都储存在PaddleVideo/output/BMN中，在模型导出步骤中可以自行选择要导出的模型。多尝试几个epoch的训练模型，择取最优者。

5）我们的方案是基于官方baseline（[使用Paddle实现乒乓球时序动作定位开源方案 - 飞桨AI Studio - 人工智能学习实训社区 (baidu.com)](https://aistudio.baidu.com/aistudio/projectdetail/3464025))的改进，该方案的数据划分是以每4s分割一次，按照1-4，4-8，8-12...依次进行；这样按照真实动作的开始时间和结束时间进行分割，能保证分割片段中都是完整的动作。

6）每一次进行推理预测前，需要将work/data/test\_npy文件夹删除，防止每次增加重复的处理后的测试数据。