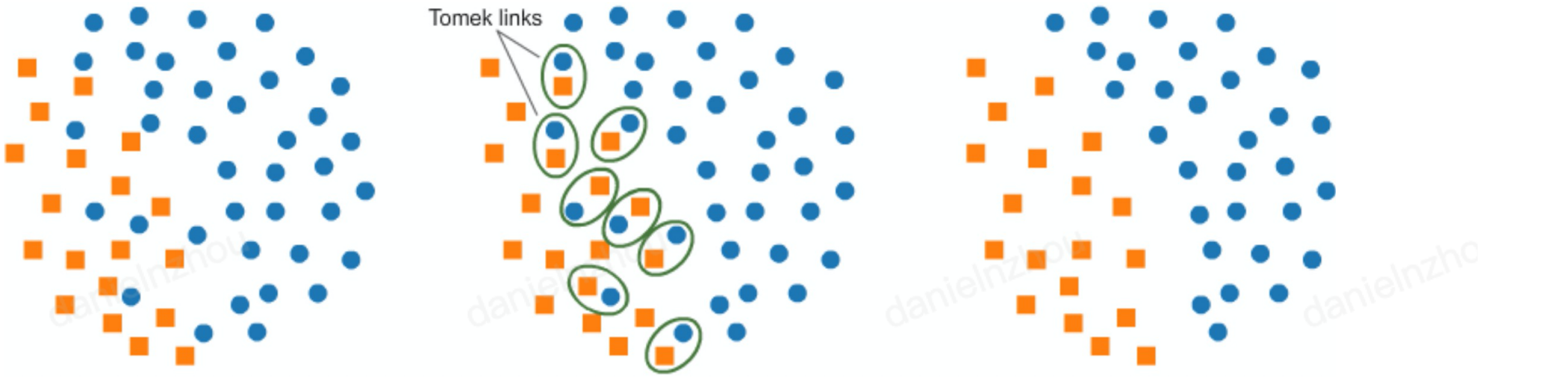


添加 Gap Interval

由 danielnzhou(周亚楠)创建, 最后修改于一月 19, 2022

1. Tomek Link Removal (TLR) 样本不均衡的做法

采用 Tomek Link Removal 的思路，其思想是，类别间的边缘可能增大分类难度，通过去除边缘中的多数类样本可以使得类别间margin更大，便于分类。

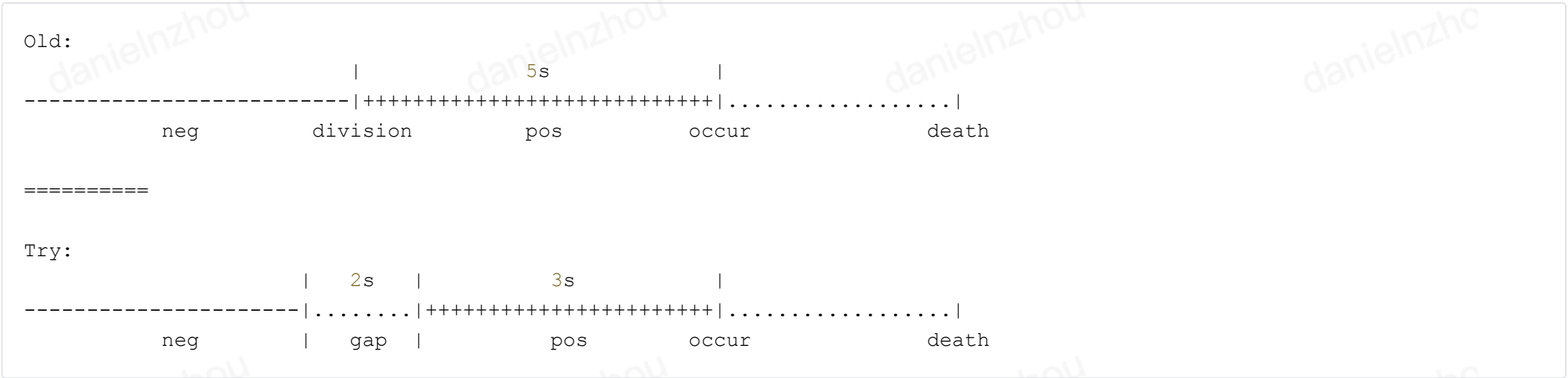


2. 意图提取事例说明

在样本提取中体现为把作为正负样本分界线的 division sample 用一个 gap interval 代替。

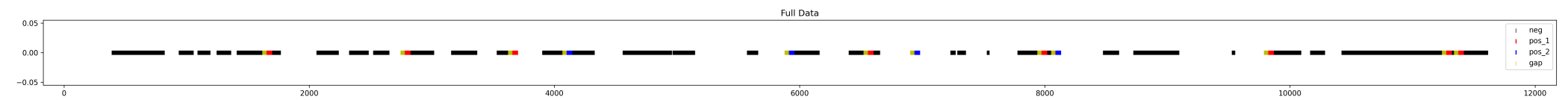
因为 division sample 附近的正负样本的区分度不是很大。

但是作为注意的是，为了提高模型的精度 precision，在测试的时候，需要把 gap interval 中间的样本点也作为正样本。



3. 时间轴每帧样本可视化

进行时间轴上的可视化，以一局游戏作为例子：

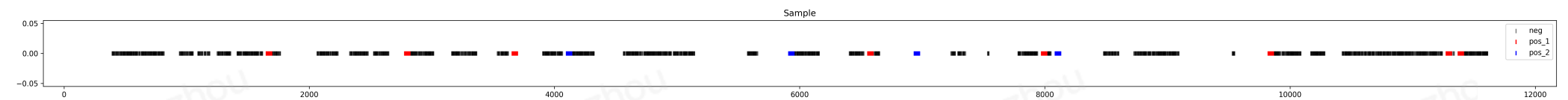


- 黑色的点是 负样本
- 红色的点是 来自 camp1 的正样本，红色的最后一帧即意图发生的帧，提取之前3S作为训练集
- 蓝色的点是 来自 camp2 的正样本，同红色
- 黄色的点是 gap interval，train 不作为样本点，eval 时可作为 pos
- 没有颜色填充的帧数，即没有团战爆发的和平时期，不作为样本点

可以看出，正样本（包括红色和蓝色）之前的2S内的帧数都被标成了黄色即 gap interval，不会作为样本放入 train data 中

因为负样本的数量远远多于正样本的数量，所以在训练的时候采用 1:8 的抽样

对负样本采用抽样的方法，黑色的点，取部分的负样本作为负样本。



4. 代码部分

```
gap interval

# s_intention_step_1.py
# positive 和 gap 保留多长时间
```

```
duration_pos_S = config["duration_pos_S"]
duration_gap_S = config["duration_gap_S"]
print(f"\nduration_pos_S: {duration_pos_S}")
print(f"duration_gap_S: {duration_gap_S}")

duration_reserve_S = duration_pos_S + duration_gap_S

duration_pos_F = duration_pos_S * FRAME_PER_SECOND
duration_reserve_F = duration_reserve_S * FRAME_PER_SECOND
```