

不会的，这个版本不会在惯性滚动后“又加速”，原因如下：

1 原理

我们用的是“滚动停止检测定时器”：

```
scrollEndTimer = setTimeout(() => {
    handleScrollEnd(x);
}, 120);
• 每次 onScroll 触发就重置定时器
• 只有 120ms 内没有 scroll 事件触发时才执行 handleScrollEnd
• handleScrollEnd 里用 scrollTo({ x: page * width,
    animated: true }) 吸附到最近页
```

重点：这里的 `animated: true` 只是让页面慢慢滑到最近页，而不是“增加速度”
浏览器惯性已经结束，ScrollView contentOffset 已经稳定了

2 为什么不会加速

- 浏览器惯性滚动只影响 scroll 事件触发阶段
- 当 120ms 无 scroll 触发 → 惯性已经结束
- 吸附动画只是把 scrollOffset 平滑滑到最近页
- 不会叠加惯性，也不会再加速滚动

3 视觉体验

阶段	scrollOffset	动画效果
手指拖动	跟手	无变化
手指松开，惯性滚动	contentOffset 自然滑动	浏览器自身惯性
惯性结束 → 120ms 无 scroll	contentOffset 稳定	scrollTo 平滑滑到最近页（很短的动画）

✓ 结果：手感平滑、没有“弹射 / 再加速”的感觉

4 可调节参数

- 定时器间隔: 100~150ms, 太小可能误触, 太大感觉慢
- 动画时长: ScrollView scrollTo 内部动画在 web 上一般默认 250ms, 可以通过 CSS 解决更精细动画
- 吸附阈值: 只在超过 1px 的偏差时吸附, 避免微抖动被触发

总结

- 这种“定时器 + scrollTo 吸附”的方案, 不会导致滚动加速
- 手指拖动顺畅、惯性自然
- WAP 上兼容性最稳定
- 可以和分页轮播、无限循环、自动轮播结合使用

如果你愿意, 我可以给你写一个完整的 WAP 轮播组件:

- 支持 水平多页
- 无限循环
- 自动播放
- 惯性自然 + 吸附平滑
- 纯 RN + WAP 兼容, 不依赖任何第三方库

我可以直接贴完整代码给你。

你希望我直接写这个吗?