

- 横竖屏切换时，ScrollView 还保留旧的像素滚动偏移，itemWidth 改变后就不再落在列边界，导致逻辑索引漂移。
- 用 ref 记住旧的 itemWidth，在布局宽度变化时用旧宽度反算出当前列索引，立即按新宽度对齐滚动位置，并同步相关的 offset/index 定位；若当前没有手势，随后再重启自动轮播。

```

const prevItemWidthRef = useRef(itemWidth);

useEffect(() => {
  if (!itemWidth || totalCount <= 1) {
    prevItemWidthRef.current = itemWidth;
    return;
  }

  const prevItemWidth = prevItemWidthRef.current;
  let targetIndex = indexRef.current;

  if (prevItemWidth && prevItemWidth !== itemWidth) {
    const approxIndex =
      Math.round(currentOffsetRef.current / prevItemWidth);
    if (!Number.isNaN(approxIndex) &&
        Number.isFinite(approxIndex)) {
      targetIndex = Math.max(0, Math.min(approxIndex,
        totalCount - 1));
    }
  }
}

prevItemWidthRef.current = itemWidth;

const nextOffset = targetIndex * itemWidth;
currentOffsetRef.current = nextOffset;
dragStartOffsetRef.current = nextOffset;
syncIndex(targetIndex);

requestAnimationFrame(() => {
  scrollToIndex(targetIndex, false);
  scheduleAdjust(targetIndex);
});

if (!isPointerActiveRef.current) {
  startAutoScroll();
}
}, [itemWidth, totalCount, scheduleAdjust,
scrollToIndex, startAutoScroll, syncIndex]);

```

建议在 Android 真机/模拟器上反复横竖屏测试，确认角标和列对齐始终保持正确。