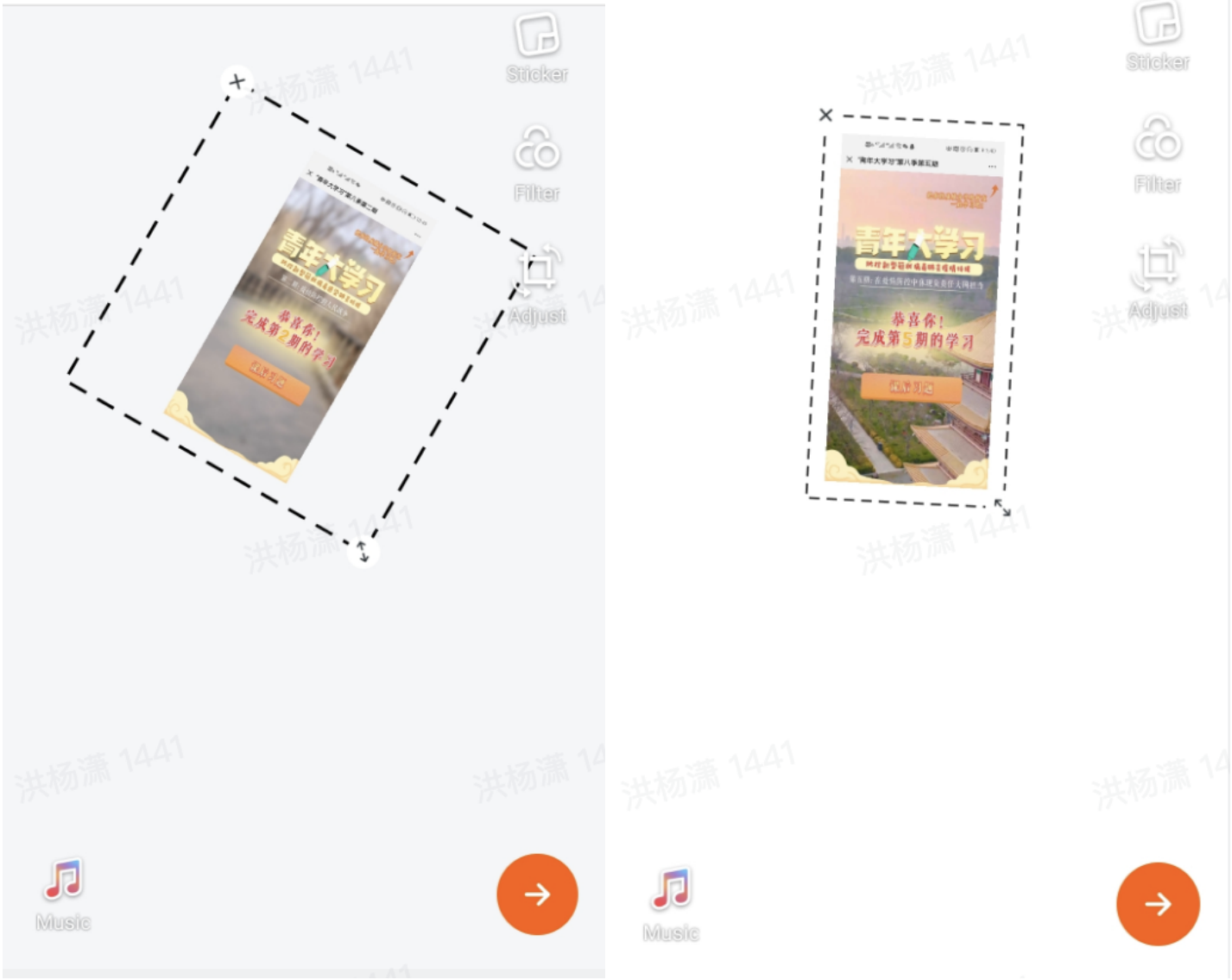
自定义水印优化

1.Watermark自定义水印边框的padding的设置

不同的图片对应同样大小的边框(写死的值)->通过图片的path获取该图片的宽和高，对图片宽和高进行计算，动态的设置当前View的width和height(通过*layoutParams*)



2.Watermark自定义水印padding的改进

当图片缩小时，图片不会完全包含在虚线框内，自定义水印padding设置的太小，导致虚线框和图片的距离过近(甚至虚线框小于图片)，设置合适的padding，并将设置View的padding的操作放在ViewControlHelper中进行。

3.不加载图片的前提下获取图片的宽和高

使用BitmapFactory.Options，并设置其inJustDecodeBounds属性为true，代替直接获取Bitmap对象。

```
/**
 * 在不加载该图片的前提下根据图片的路径获取该图片的长度和高度
 * imagePath:图片的路径
 * return:返回图片的长度和高度，保存在数组中
 */
private fun getImageWidthAndHeight(imagePath:String):IntArray{
    val options = BitmapFactory.Options()
    options.inJustDecodeBounds = true
    // 可通过
    val bitmap :Bitmap! = BitmapFactory.decodeFile(imagePath,options)
    return intArrayOf(options.outWidth,options.outHeight)
}
```

```
val image : Bitmap! = BitmapFactory.decodeFile(bean.imagePath)

val imageWidth : Int = image.width
val imageHeight : Int = image.height
```