Unity中支持两种批处理方式：一种是动态批处理，另一种是静态批处理。对于动态批处理来说，优点是一切处理都是unity自动完成的，不需要我们自己做任何操作，而且物体是可以移动的。但缺点是，限制很多，可能一不小心就会破坏了这种机制，导致unity无法动态批处理一些使用了相同材质的物体。而对于静态批处理来说，它的有点是自由度很高，限制很少，但缺点是可能会占用更多的内存，而且惊涛静态批处理后的所有物体都不可以再移动（即便在脚本中尝试改变物体的位置也是无效的）

**动态批处理**

如果场景中有一些模型共享了同一个材质并满足一些条件，unity就可以自动把他们进行批处理，从而只需要花费一个drawcall就可以渲染所有的模型。动态批处理的基本原理是，每一帧把可以进行批处理的模型网格进行合并，再把合并后模型数据传递给GPU，然后使用同一个材质对其渲染。除了实现方便，动态批处理的另一个好处是，经过批处理的物体仍然可以移动，这是由于在处理每一帧时Unity、都会重新合并一次网格。

虽然Unity的动态批处理不需要我们进行任何额外工作，但只有满足条件的模型和材质才可以被动态批处理。需要注意的是，随着Unity版本的变化，这些条件也有一些改变，这里给出了一些主要的条件限制：

1. 能够进行动态批处理的网格的顶点规模要小于900。例如，如果shader中需要使用顶点位置、法线和纹理坐标这3个顶点属性，那么想要让模型能够被动态批处理，它的顶点数数目不能超过300。需要注意的是，这个数字在未来有可能会发生变化，因此不要依赖这个数据
2. 一般来说，所有对象都需要使用同一个缩放尺寸（可以是（1，1,1）、（1,2,3）、（1.5,1.4,1.3）等，但都必须一样）。一个例外情况是，如果所有的物体都使用了不同的非统一缩放，那么他们也是可以被动态批处理的。但在Unity5中，这种对模型缩放的限制已经不存在了
3. 使用光照纹理（lightmap）的物体需要小心处理。这些物体需要额外的渲染参数，例如，在光照纹理上的索引、偏移量和缩放信息等。因此，为了让这些物体可以被动态批处理，我们需要保证他们指向光照纹理中的同一个位置
4. 多Pass的shader会中断批处理。在前向渲染中，我们有时需要使用额外的Pass来为模型添加更多的光照效果，但这样一来模型就不会被动态批处理了。