Suppose that we design a deep architecture to represent a sequence by stacking self-attention layers with positional encoding. What could be issues?

Several issues could arise by stacking self-attention layers with positional encoding. First, output of the previous attention layers carries residual positional information into the following layers. Thus, it could be argued that positional encoding is not needed in every attention layers. This setup can cause additional unnecessary time of training, speed of inferencing, and space to allocate.

Second, different positional encoding methods may inherently carry with them different issues. Absolute positional encodings are simple yet are not able to express relative positional information. Relative positional encodings can address such issue but also come at the cost of greater time and space complexities. Finally, learnable positional encoders are the most robust method. However, it is even more costly to compute and harder to train. In addition, due to its complexity, it is almost not interpretable.

Third, simply stacking attention layers may cause tremendous challenges to hardware, since it has tremendously more parameters than that of a fully connected or convolutional layers. In addition, it is not necessarily the case that the represented sequence needs to be expressed by such a complex model. It is always important to gauge the scale of the problem. Modifying the architecture to fewer layers of attentions and more MLPs may be more effective. Lastly, stacking attention layers irresponsibly can also come at the challenge of overfitting the sequence. In such a case, it is also worthwhile considering reducing the complexity of the models or adding regularizations.